TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABREVIATIONS

DCS	Dynamic Classifier Selection
DES	Dynamic Ensemble Selection
DSEL	Dynamic Selection Dataset
DSLP	Dynamic Selection Local Pool
DSPG	Dynamic Selection Pool Generation
ÉTS	École de Technologie Supérieure
KNN	Dynamic Selection Dataset
RoC	Region of Competence
SFR	Number of sample on the number of features ratio
TP	Temporary Pool

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

c_i	i-th classifier
C	Pool of classifiers
C^*	Ensemble of most competent classifiers
c^*	The most competent classifier
Ω	Set of class labels of x_k
w_k	Class label
L	Number of class labels
N	Number of instances in a dataset
T_r	Training dataset
T_e	Testing dataset
V_a	Validation dataset
θ	Region of competence
x_q	test query
x_j	j-th sample
$\delta_{i,j}$	Level of competence of the classifier c_i for classifying the instance x_q

INTRODUCTION

Through the decades, various research projects have been conducted in the area of classification in a wide range of sectors. Despite the innumerable classification methods that cover different aspects of classification problems, empirical and theoretical results drove the researchers to jointly agree that building a single robust classifier isn't always the right fit to deal with all the complex pattern recognition problems (Britto *et al.*, 2014). Therefore, it took years to the computational intelligence community to converge to a research trend that focuses on Ensembles of classifiers (EoC) or commonly called Multiple Classifier Systems (MCS).

Multiple Classifier Systems tend to mimic the human nature that usually seeks for different opinions before making a final decision (Rokach, 2010). Researchers from diverse disciplines such as pattern recognition, statistics, and machine learning have explored the use of ensemble methods since the late seventies (Rokach, 2010). They have been acknowledged for their conceptual simplicity and the top-level performance in many classification tasks (Tamponi, 2015; Friedman *et al.*, 2001). The process of MCS is constituted of three major phases: generation, selection and integration (Britto *et al.*, 2014). As shown in Figure 0.1, during the first phase, a pool of classifiers is generated; in the second step, one classifier or a non-empty subset of these classifiers is selected, while in the last one, a final decision is made based on the prediction(s)/opinion(s) of the selected classifier(s) (Britto *et al.*, 2014).

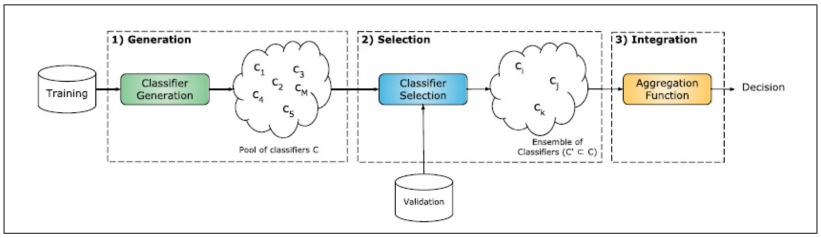


Figure 0.1 The three phases of a MCS system: pool generation, classifiers selection and integration adapted from (Cruz *et al.*, 2018)

Among these phases, intensive work on the selection phase became a key cause for the evolution of MCS. Selection can be achieved in two ways, static and dynamic. In the Static Selection approach, the subset of classifiers is determined during the training of MCS; therefore, the same subset is applied for all test samples. On the contrary, Dynamic Selection (DS) considers different subsets of classifiers for individual test samples (Roy *et al.*, 2016). Several techniques have been developed through the years (Ko *et al.*, 2008; Cruz, 2016) aiming to enhance the strength of the recognition rates. However, there is no single algorithm that is better than any other over all possible classes of problems as stated in the "No free lunch" theorem (Cruz, 2016; Cruz *et al.*, 2018; Britto *et al.*, 2014).

Problem statement

In the traditional process of Multiple Classifier Systems for DS, the phases consist on generating a diverse pool of classifiers with mass generation techniques such as Bagging, Boosting, Random Subspace etc (Britto *et al.*, 2014; Cruz *et al.*, 2018). Then comes the step where dynamic selection techniques select a classifier or a subset of classifiers, for each test sample.

The issue with these generation approaches is that they were designed for static combination methods. In other words, they use a global approach in generating the base classifiers (Souza *et al.*, 2017; Cruz *et al.*, 2018).

Although, the results of this process are efficient enough through the advancement of the field, the performance in generalization for DS depends on the initial base classifiers of the pool. Furthermore, the main philosophy of dynamic selection, is to select the most competent classifiers locally.

However, with ensembles generated with the static combination methods, this condition is not taken into consideration. It means that there is no guaranty for the the presence of local experts, which leads to the incapacity of the DS technique to always select local competent classifiers (Cruz *et al.*, 2018; Oliveira *et al.*, 2017).

Previous work in the literature, focused on the elaboration of several new Dynamic Selection methods, by creating new competence measures, and new frameworks to improve the performances(Lustosa Filho *et al.*, 2018; Oliveira, 2018). Recently, Cruz *et al.*, (Cruz *et al.*, 2018), addressed the issue of rethinking the generation phase of these base classifiers, before applying the DS techniques.

To the best of our knowledge, there exists no pool generation method that uses local information to suit dynamic selection, in its classical scheme (Cruz *et al.*, 2018; Oliveira, 2018), apart from an online generation of classifiers conducted recently by souza *et al* in (Souza *et al.*, 2019).

Therefore, this work aims to provide an approach leading towards local classifier generation for dynamic selection, by taking into consideration local criteria within the creation of the classifiers. Which, we believe is a promising subject to explore.

How can we generate locally competent classifiers that are adapted to the Dynamic Selection scheme?

To achieve the creation of locally competent classifiers we believe that the following criteria must be met:

- The presence of at least one classifier that crosses the Region of Competence of the patterns located in indecision regions.
- The use of different guides borrowed from the Dynamic Selection scheme for pool generation.

Research goals and Contributions

Therefore, our research question leads us to propose the following :

1. A novel procedure to generate local classifiers that cross the region of competence (RoC). The method takes into account the different local information regarding the samples that constitute RoC.
2. A heuristic to gather the previously generated classifiers given different types of guides for the construction of the Dynamic Selection Local Pool (DSLP), that will be detailed in chapter 2.
3. A classification system that takes in considerations the hardness of its test samples to decide whether to use the KNN classifier and Dynamic Selection techniques based on the recommendations provided in our previous work (Cruz *et al.*, 2017).

Pursuing the research goal to rethink the usual pool generations methods by including information, holds promise towards the generation of local classifiers for Dynamic Selection.

Organization of the thesis

This document is organized as shows the figure 0.2: chapter one presents the related work about Dynamic Selection and Ensemble generation methods. The second chapter describes the proposed system heading towards local pool generation for Dynamic Selection, we presented in the orange boxes the titles of the most important aspects of the chapters. We expose in chapter three, the experimental protocol, as well as the results, the comparison to the state of the art and their discussion. The related work and the other chapters are linked with a dashed line to Appendix I, representing a conference paper that performs an analysis of the performances of the Dynamic Selection scheme and the K-NN classifier. This is a complementary reading that supports our research direction. Finally, we conclude by summarizing this study and give recommendations for future work.

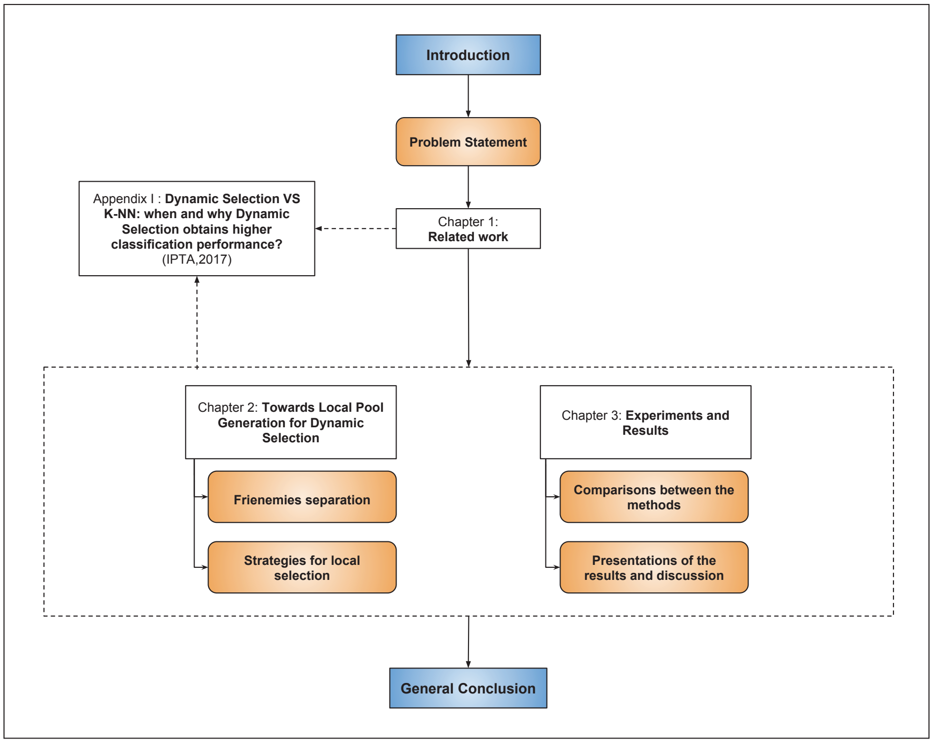

Introduction
Problem Statement
Appendix I : Dynamic Selection VS K-NN: when and why Dynamic Selection obtains higher classification performance? (IPTA,2017)
Chapter 1: Related work
Chapter 2: Towards Local Pool Generation for Dynamic Selection
Frienemies separation
Strategies for local selection
Chapter 3: Experiments and Results
Comparisons between the methods
Presentations of the results and discussion
General Conclusion

Figure 0.2 Thesis plan

CHAPTER 1

RELATED WORK

This chapter regroups the literature review concerning the Oracle as an important concept in the Multiple Classifiers Systems. It then presents the Dynamic selection scheme and the Ensemble generation methods. It also contains complementary sections that narrows down the understanding of the problem and introduces the motivations of the solutions proposed in the next chapter.

1.1 The Oracle

In the process of the ideal selection of classifiers, the concept of the Oracle is defined as an abstract function that always chooses the classifier that predicts the correct label for each instance (Kuncheva, 2002, 2004b; Cruz *et al.*, 2018), if there exists such a classifier. That is to say, it represents the ideal classifier selection scheme. For a given dataset, the classifier c_i gives an output vector y_j (Kuncheva, 2004b) so that :

$$y_{i,j} = \begin{cases} 1 & \text{if classifier } c_i \text{ correctly classifies } x_j \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

In the area of Multiple Classifiers Systems, the concept of the Oracle is exploited at different stages. It is used to the construction of diverse pools (Kuncheva, 2004b). Diversity is agreed that it measures how complementary the classifiers are in terms of making different mistakes in the features space (Kuncheva, 2004b). Kuncheva (Kuncheva, 2004b) states the following concerning the diversity concept: '*If we have a perfect classifier which makes no errors, then we do not need an ensemble. However,if the classifier does make errors, then we seek to complement it with another classifier which makes errors on different objects. The diversity of the classifier outputs is therefore a vital requirement for the success of the ensemble*'. Moreover, empirical results showed that there exists a positive correlation between accuracy of the ensemble and diversity among the base classifiers, which leads to a high accuracy of the Oracle when the diversity rate is high (Kuncheva, 2004b; Shipp & Kuncheva, 2002; Tang *et al.*, 2006).

On the other hand, in the context of the Dynamic Selection literature, the Oracle is used to determine whether the results obtained by dynamic selection techniques is close to the ideal accuracy (Cruz *et al.*, 2018; Souza *et al.*, 2017; Giacinto & Roli, 2001). In other words, it acts as an upper bound for DS techniques performances for a given pool of classifiers. It also contributes in the elaboration of many ensemble generation methods (Souza *et al.*, 2017; Dos Santos *et al.*, 2008; Santos & Sabourin, 2011; Kuncheva & Rodriguez, 2007) as well as Dynamic selection techniques and frameworks (Cruz, 2016; Ko *et al.*, 2008; Oliveira, 2018).

Given the quick introduction about the Oracle as an important element in the context of MCS, the next sections present the Dynamic Selection concept followed by the Ensemble generation scheme as a solid background to support the research question concerning the creation of locally competent classifiers for Dynamic Selection.

1.1.1 Dynamic Selection

In the context of MCS, the selection of classifiers can be either static or dynamic (Britto *et al.*, 2014; Cruz *et al.*, 2018). The former considers a subset of base classifiers for all the test patterns whereas the latter assumes that each classifier is an expert in a specific region of the features space (Britto *et al.*, 2014; Cruz *et al.*, 2015b; Cruz, 2016; Cruz *et al.*, 2018). Therefore, each query instance is classified by a single classifier or an ensemble of classifiers. Empirical studies showed that Dynamic Selection (DS) is well suited for ill defined problems, i.e., for small sized datasets and when there is insufficient training data (Britto *et al.*, 2014; Cruz *et al.*, 2015b; Cruz, 2016; Cruz *et al.*, 2018). Moreover, several research projects have been focusing on Dynamic Selection, which can be applied in several domains such as : music genre classification, credit scoring, face recognition, signature verification, bug predictions and many more (Cruz *et al.*, 2018).

Based on a pool of supposedly diverse classifiers $C = \{c_1, ..., c_M\}$, dynamic selection consists on finding the most competent classifier or ensemble of classifiers $C \subset C$ to predict the class label for each test sample x_q (Kuncheva, 2004b). In Dynamic Selection, the classification of a new test samples unfolds the following steps:

1. The definition of a local region in which the selection will operate called, Region of Competence (RoC).
2. The selection criterion used to estimated the competence of the base classifiers.
3. The selection scheme. Dynamic selection techniques can be divided into two categories: Dynamic Classifier Selection (DCS) or Dynamic Ensemble Selection (DES). In dynamic classifier selection, a single classifier is selected for each test sample whereas in dynamic ensemble selection, a subset of competent classifiers is selected for each test pattern.

Figure 1.1 shows a diagram of the Taxonomy of DS adapted from (Cruz *et al.*, 2018). It shows the different processes of Dynamic Selection, enumerating the region of competence definition, the selection criteria (individual-based and group-based measures of competence) and the Selection Approach.

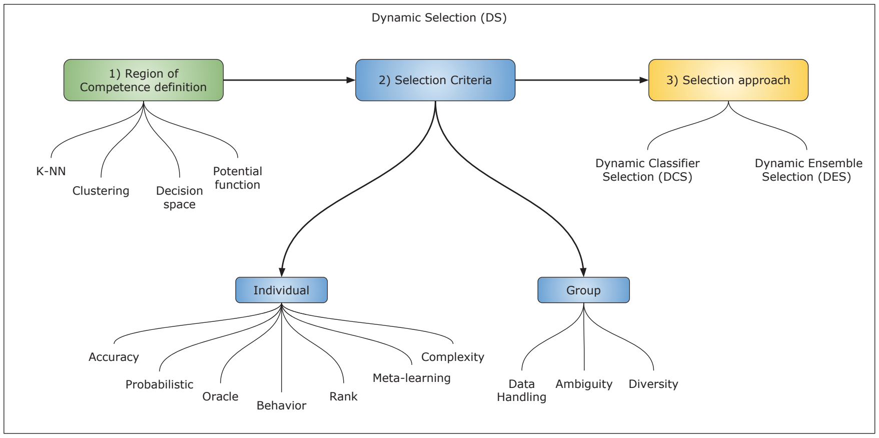
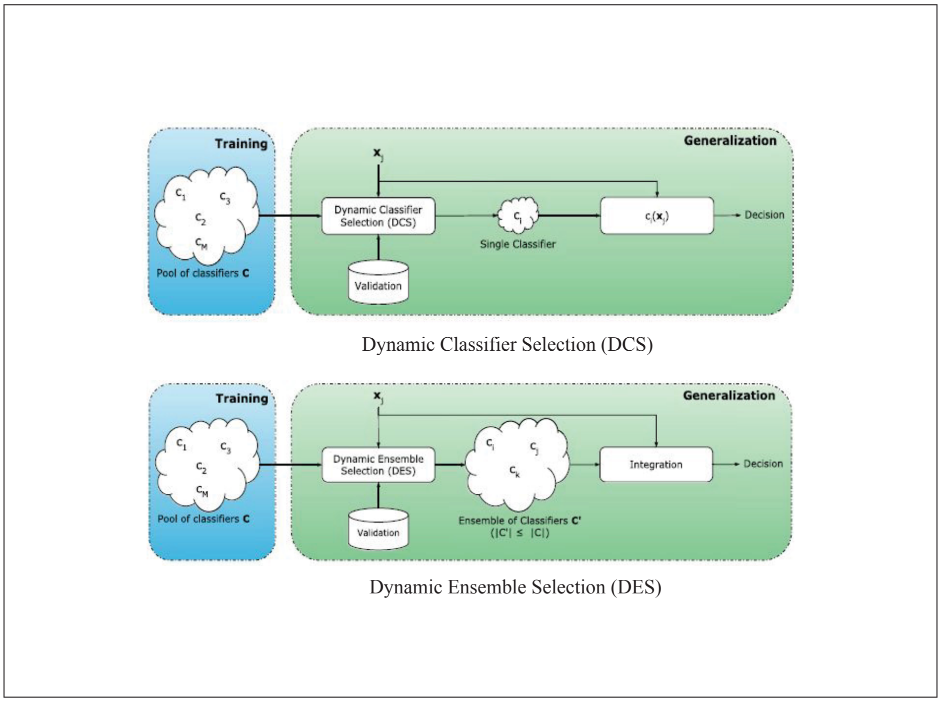


Figure 1.1 Taxonomy of the Dynamic Selection Scheme (Cruz *et al.*, 2018)

Figure 1.2 shows a basic dynamic classifier selection system differentiating between the Dynamic Classifiers Selection (DCS) and Dynamic Ensemble Selection (DES). Indeed, the prior uses only one classifier defined as the most competent one to identify the test sample whereas the latter uses an ensemble of competent classifier for the recognition task (Britto *et al.*, 2014; Cruz *et al.*, 2018).

In this section, we address the different concepts used in the elaboration of the Dynamic Selection systems, we will first give a definition of the local regions where the decisions of the system occurs. Then, we expose the measures of competence evaluated for the DS techniques and introduce them in the same subsection. Furthermore, we exhibit complementary information about the behavior of the DS scheme in certain situations to provide a broad motivation to our research question.



Dynamic Classifier Selection (DCS)

Dynamic Ensemble Selection (DES)

Figure 1.2 Classical Dynamic Selection System showing the difference between the Dynamic Classifier Selection (DCS) and Dynamic Ensemble Selection (DES) adapted from (Cruz *et al.*, 2018)

1.1.2 Region of Competence definition

The Region of Competence in the Dynamic Selection scheme represents the local region of the features space that encloses the query x_q, it is also where the competence of the classifiers is estimated to recognize x_q. The DS techniques are very sensitive to the definition of the Region of Competence that is composed of labeled samples usually from the validation set that is called the Dynamic Selection Dataset *(DSEL)* (Cruz *et al.*, 2018). The local region for several DS techniques is defined using the K-Nearest Neighbors rule, a clustering technique, a potential function, or a decision space scheme and strongly depends on the distribution of DSEL (Cruz *et al.*, 2018; Didaci & Giacinto, 2004; Cruz *et al.*, 2015a).

Clustering

To set the region of competence, the clusters are defined in *DSEL* by one of the clustering techniques such as the K-Means algorithm (Kaufman & Rousseeuw, 2009). Then, for all the clusters, we estimate the local competence of each base classifier (Cruz *et al.*, 2018). During the testing phase, given a new query sample, x_q, we calculate the distance between this example and the centroid of each cluster. The competence of the classifiers is accessed according to the examples that belong to the closest cluster (Cruz *et al.*, 2018).

K-Nearest Neighbors

The K-Nearest Neighbor approach is conducted within *DSEL* for estimating the K nearest neighbors for a test query x_q. The region of competence θ is then defined and the decisions taken by the DS corresponding DS techniques are based on the elements of θ (Cruz *et al.*, 2018).

Potential Function

The particularity of the methods based on the potential functions model resides in the use of the whole *DSEL* for computing the region of competence instead of the neighborhood (Cruz *et al.*, 2018). Each sample $x_i \in DSEL$ is weighted using its Euclidean distance (d) to the query x_q using usually, a Gaussian potential function model K (Equation 1.3) (Cruz *et al.*, 2018). The function gives the highest weights to samples nearest to the test sample, and lowest weights to samples distant from the test sample. As a consequence, the data points that are the closest to x_q have a higher impact on the classifiers' competence estimation(Cruz *et al.*, 2018).

$$K(x_i, x_{query}) = exp(-d(x_i, x_{query})^2) \tag{1.2}$$

Decision Space

This category focuses on the behavior of the classifiers and the use of their predictions as information sources. The estimation of the region of competence is conducted using the decision spaces instead of the features space inspired by Behavior Knowledge Space (BKS) (Huang & Suen, 1995). This approach transforms the test sample x_q and $DSEL$ into output profiles; which are vectors composed of predictions of the base classifiers in the pool either using the hard decisions (Huang & Suen, 1995) or exploiting the estimated posterior probabilities of the classifiers, as stated in (Cavalin *et al.*, 2013, 2012; Batista *et al.*, 2011; Oliveira, 2018; Cruz *et al.*, 2018). The selection of the samples that compose the region of competence is given by the points in $DSEL$ with the most similar output profiles to the output profile of x_q.

1.1.3 Measures of competence and Dynamic Selection techniques

Classifier competence defines how much we trust an expert given a classification task (Cruz *et al.*, 2018; Britto *et al.*, 2014). The level of competence of each base classifier is measured taking into account each new test instance, and only the classifiers that reach a certain level of competence for the current test instance are selected to compose the ensemble (DES) or the classifier (DCS). On the other hand, it is necessary to define the criterion to measure the level of competence of each classifier. For dynamic selection techniques, the search criterion is locally applied to fit each test pattern (Cruz *et al.*, 2018).

There are two types of dynamic selection criteria, individual-based level of competences known as: Ranking, Local Accuracy, Oracle, Probabilistic, Behavior and group-based: Diversity, Ambiguity and Data handling (Britto *et al.*, 2014; Cruz *et al.*, 2015b; Cruz, 2016; Cruz *et al.*, 2018).

For the definitions, $\theta_{x_q} = \{x_1, ..., x_k\}$ represents patterns belonging to the region of competence of the unseen sample x_q, K is the size of the RoC, c_i is the base classifier from the pool C, w_k is the class attribute of x_k and $\delta_{i,q}$ is the level of competence of the classifier c_i for classifying the instance x_q .

1.1.3.1 Individual-based measures of competence

Ranking

Several Dynamic Selection techniques have been developed according to this taxonomy. For the individual-based measures, **Classifier Rank** (Sabourin *et al.*, 1993; Cruz *et al.*, 2018; Britto *et al.*, 2014) is considered as one of the first proposed approaches to estimate the base classifiers' competence level in DS. The ranking of a classifier c_i is found by counting the number of consecutive correctly classified samples (Cruz *et al.*, 2018). The classifier that correctly classifies the most consecutive samples coming from the region of competence is considered to have the highest competence level (Cruz, 2016).

Accuracy

Overall local accuracy (OLA) estimates each individual classifier's accuracy in local regions of the feature space surrounding a test sample, and then uses the decision of the most locally accurate classifier (Woods *et al.*, 1997; Britto *et al.*, 2014; Cruz *et al.*, 2018). The level of competence $\delta_{i,q}$ of a base classifier c_i is computed as the percentage of samples in the region of competence that are correctly classified.

$$\delta_{i,q} = \frac{1}{K}\sum_{k=1}^{K} P(w_l\,|\,x_k{\in}w_l, c_i) \tag{1.3}$$

Local class accuracy (LCA) is similar to **OLA**, the only difference being that the local accuracy is estimated in respect of output classes w_l (w_l is the class assigned to the query by c_i) (Woods *et al.*, 1997; Britto *et al.*, 2014; Cruz *et al.*, 2018) for the whole region of competence.

$$\delta_{i,q} = \frac{\sum_{x_k \in w_l} P(w_l\,|\,x_k, c_i)}{\sum_{k=1}^{K} \; P(w_l\,|\,x_k, c_i)} \tag{1.4}$$

Modified local Accuracy (MLA) works similarly to the **LCA** technique, with the only difference being that each instance in the region of competence is weighted by its Euclidean distance W_k to the query instance. That way, instances from the region of competence that are closer to the test sample have a higher influence when computing the performance of the base classifier (Smits, 2002; Britto *et al.*, 2014; Cruz *et al.*, 2018).

$$\delta_{i,q} = \sum_{k=1}^{K} P(w_l\,|\,x_k{\in}w_l, c_i)\,W_k \tag{1.5}$$

Probabilistic

In the **probabilistic measures**, two methods: **A priori** and **A posteriori** are “evolved” versions of **OLA**. The prior selects a single classifier from the pool based on a local region defined by the K-nearest neighbors of the test pattern in the training set during the testing phase without considering the class assigned to the unknown pattern (Britto *et al.*, 2014; Giacinto & Roli, 1999; Cruz *et al.*, 2018). This measure of classifier accuracy is calculated as the class posterior probability of the classifier c_j on the neighborhood of the unknown sample.

$$\delta_{i,q} = \frac{\sum_{k=1}^{K} P(w_l\,|\,x_k{\in}w_l,\, c_i)W_k}{\sum_{k=1}^{K} W_k} \tag{1.6}$$

Similarly, the latter estimates local accuracies using the class posterior probabilities and the distances of the samples in the defined local region (Britto *et al.*, 2014; Giacinto & Roli, 1999; Cruz *et al.*, 2018).

$$\delta_{i,q} = \frac{\sum_{x_k \in \omega_l} P(w_l \,|\, x_k \in w_k,\, c_i) W_k}{\sum_{k=1}^{K} P(w_l \,|\, x_k \in w_k,\, c_i) W_k} \tag{1.7}$$

Behavior

Giancinto and Roli (Giacinto & Roli, 2001) proposed the **Multiple Classifier Behavior (MCB)** algorithm that is a mixture of the Dynamic Classifier Selection Local Accuracy with the behavior-knowledge space **(BKS)** (Huang & Suen, 1995). It is based on a similarity function that measures the proportion of similarities between all the output profiles of the classifiers (Huang & Suen, 1995; Cruz *et al.*, 2018; Britto *et al.*, 2014). The method defines the region of competence θ_{x_q} using the K-NN method. The similarity function acts as a behavioral similarity detector that preselects from θ_{x_q} from which the classifiers showed similar behavior to the one observed for the unknown pattern x_q (Britto *et al.*, 2014).The rest of the instances are exploited to select the most accurate classifier by using the OLA. At last, based on a predefined threshold, if the selected classifier outperforms the others in the pool; it is used to classify the unknown sample, if not, all the classifiers in the pool will perform the classification of x_q (Oliveira, 2018; Cruz *et al.*, 2018; Britto *et al.*, 2014).

The similarity function is built as follows:

$$similarity(A,B) = \frac{1}{M} \times \sum_{j=1} MT(A_j, B_j) \tag{1.8}$$

where A and B are vectors, M is the size of the vectors A and B, and T is the $XNOR$ function (1 when $A_j = B_j$, otherwise 0).

Oracle

Ko et al. (Ko *et al.*, 2008) passed from Dynamic Classifier Selection **(DCS)** to Dynamic Ensemble Selection **(DES)** by developing the concept of the **K-nearest-Oracles (KNORA)**, which is close to the concepts of OLA, LCA, the A Priori and A Posteriori methods in their consideration of the neighborhood of test patterns, but it can be distinguished from the others by the direct use of the oracle property of having training samples in the region of competence with which to find the most suitable ensemble for a given query (Ko *et al.*, 2008). For any test data point, **KNORA** simply finds its nearest K neighbors in *DSEL*, assess which classifiers correctly classify those neighbors and uses them as the ensemble for classifying the given pattern in that test set(Ko *et al.*, 2008). **KNORA** has different designs, we can state the following ones:

KNORA-ELIMINATE (KNORA-E) The KNORA-Eliminate approach exploits the concept of the Oracle (Ko *et al.*, 2008) which is the upper limit of a classifiers ensemble. For a region of competence θ_q of a given query x_q in *DSEL*, sole the classifiers that correctly classify all the neighborhood samples (achieving a 100% accuracy, hence, operating as "local oracles") are selected(Ko *et al.*, 2008) to build the ensemble. The selected base classifiers' decision is combined using majority voting. In the case where, there exists no classifiers that perfectly classify all the neighborhood samples, the method reduces the size of θ_q by eliminating the samples that are most distant from the x_q until at least one classifier is chosen.

KNORA-E has another alternative called "KNORA-E-W" which is a weighted version of the original KNORA-E, according to the Euclidean distance between the samples in DSEL and the test query (Ko *et al.*, 2008).

KNORA-UNION (KNORA-U) In this scheme, KNORA-UNION operates by selecting all the base classifiers that can correctly classify at least one sample from the neighborhood θ_q.

The method grants a vote to each classifier c_i that correctly classifies one sample from the neighborhood θ_q. This means that the base classifier c_i could have more than one vote if it correctly classifies more than one sample. Therefore, the votes gathered by all the classifiers are aggregated using a majority voting rule to obtain the ensemble decision (Ko *et al.*, 2008).

KNORA-U has another alternative called "KNORA-U-W" which is a weighted version of the original KNORA-E, according to the Euclidean distance between the samples in DSEL and the test query (Ko *et al.*, 2008).

Note

Recently, Oliveira *el al.* (Oliveira *et al.*, 2018) proposed two new variants of the KNORA-E DES technique scheme. KNORA-B, B stands for borderline is a DES technique-based adapted from KNORA-E. It actually diminishes from the the region of competence but keeps at least one sample from each class that is in the original region of competence as opposed to the Original KNORA-E. KNORA-BI is a spin off of KNORA-B where I stands for imbalance datasets, which reduces the region of competence by only removing samples belonging to the majority class, leaving the minority untouched.

Meta Learning

Last but not least, **META-DES** is a recent dynamic selection framework using meta-learning (Cruz *et al.*, 2015a). *Cruz et al.* proposed five sets of meta-features to measure the level of competence of a classifier for the classification of input samples (Cruz *et al.*, 2015a). The meta-features are used to train a meta-classifier to predict whether or not a classifier is competent enough to classify an input instance. On the other hand, Cruz *et al.*, have proposed an improvement variant to the **META-DES** framework called **META-DES.Oracle**, presented in (Cruz *et al.*, 2018; Cruz, 2016) which applies a meta-feature selection scheme using Binary Particle Swarm Optimization (BPSO) to optimize the performance of the meta-classifier.

1.1.3.2 Group-based measures of competence

Group-based methods work by estimating the competence level of a whole ensemble of classifiers rather than each classifier individually.

Diversity

Diversity in the context of dynamic selection has been used by some authors as a post-processing means of improving classification performance after an ensemble is selected. Several metrics for measuring diversity in an EoC have been proposed (Cruz, 2016). Of all diversity measures, the Double-Fault (Shipp & Kuncheva, 2002) measure received a lot of interest as it presents a higher correlation with the majority voting accuracy (Shipp & Kuncheva, 2002) when compared to other diversity measures.

Ambiguity

The second group-based measure is **Ambiguity** in Dynamic Selection, there are several ways of measuring the ambiguity. As for Ambiguity-guided dynamic selection **(ADS)**, it is measured by the number of classifiers that disagree with the result of the majority vote over the ensemble (Cruz *et al.*, 2018; Britto *et al.*, 2014), the most competent ensemble is the one that produces the lowest ambiguity value.

Data handling

Data handling is an interesting adaptive ensemble selection approach based on data handling theory (GDMH, family of inductive algorithms for computer-based mathematical modeling of multi-parametric datasets that features fully automatic structural and parametric optimization of models (Ivakhnenko, 1970)) and complexity models was proposed in (Xiao & He, 2009). The system is based on a multivariate analysis theory for modeling complexity systems presented in (Ivakhnenko, 1970). Given a new test sample x_j, several ensemble configurations are evaluated using the GMDH. Then, the ensemble with optimal complexity is selected (Cruz, 2016).

Summary of the previously presented DS techniques

Table 1.1 is a brief summary holding the Dynamic Selection techniques, the way the region of competence is defined, their selection criteria and the authors who elaborated these methods organized in a chronological order. The next subsections provide a complementary discussion about some particularities of the DS scheme that are useful for the elaboration of our proposed system.

Table 1.1 A summary of the the Dynamic Selection techniques and their characteristics in a chronological order inspired from (Cruz *et al.*, 2018)

Technique	**RoC definition**	**Selection criteria**	**Selection approach**	**Reference**	**Year**
Classifier Rank (DCS-Rank)	K-NN	Ranking	DCS	Sabourin et al. (Sabourin *et al.*, 1993)	1993
Overall Local Accuracy (OLA)	K-NN	Accuracy	DCS	Woods et al. (Woods *et al.*, 1997)	1997
Local class Accuracy (LCA)	K-NN	Accuracy	DCS	Woods et al. (Woods *et al.*, 1997)	1997
Apriori	K-NN	Probabilistic	DCS	Giacinto et al. (Giacinto & Roli, 1999)	1999
Aposteriori	K-NN	Probabilistic	DCS	Giacinto et al.(Giacinto & Roli, 1999)	1999
Multiple Classifier Behavior (MCB)	K-NN	Behavior	DCS	Giacinto et al. (Giacinto & Roli, 2001)	2001
Modified Local Accuracy (MLA)	K-NN	Accuracy	DCS	P.C. Smits (Smits, 2002)	2002
K-Nearest Oracles Eliminate (KNORA-E)	K-NN	Oracle	DES	Ko et al. (Ko *et al.*, 2008)	2008
K-Nearest Oracles Union (KNORA-U)	K-NN	Oracle	DES	Ko et al. (Ko *et al.*, 2008)	2008
META-DES	K-NN	Meta-Learning	DES	Cruz et al. (Cruz *et al.*, 2015a)	2015
META-DES.Oracle	K-NN	Meta-Learning	DES	Cruz et al. (Cruz, 2016)	2016
K-Nearest Oracles Borderline (KNORA-B)	K-NN	Oracle	DES	Oliveira et al. (Oliveira *et al.*, 2018)	2018
K-Nearest Oracles Borderline Imbalance (KNORA-BI)	K-NN	Oracle	DES	Oliveira et al. (Oliveira *et al.*, 2018)	2018

1.1.4 Dynamic Selection Versus K-NN

For most DS techniques, the competence of the base classifiers are heavily dependent on the K-Nearest Neighbors for the definition of the local regions (Cruz *et al.*, 2017). Therefore, one question arose: why do we use dynamic selection instead of simply applying the K-NN classifier?

In order to answer that question, *Cruz et al.* performed an analysis comparing the classification results of DS techniques and the K-NN classifier under different conditions (Appendix I). Experiments were conducted on 18 state-of-the-art DS techniques over 30 classification datasets and results showed that DS methods present a significant boost in classification accuracy even though they use the same neighborhood as the K-NN (Cruz *et al.*, 2017). The reasons behind the out-performance of DS techniques over the K-NN classifier reside in the fact that DS techniques can deal with samples with a high degree of instance hardness (samples that are located close to the decision border) as opposed to the K-NN (Cruz *et al.*, 2017).[1]

The conclusion of this work gave a new perspective to the dynamic selection scheme. Indeed, for future work dealing with dynamic selection, they suggest a system that operates in two phases: first, the hardness of a test instance is calculated, then based on the results, the system could select whether using K-NN or applying a DS technique for classification (Cruz *et al.*, 2017). The reason behind such a choice is that, DS scheme would be only used to classify samples associated with a high degree of instance hardness i.e. borderline samples, while $K-NN$ would be used for classifying samples with a low degree of instance hardness i.e. safe samples (Cruz *et al.*, 2017). Therefore, we take this suggestion into account while building our pool generation method.

[1] This work was presented in the International Conference on Image Processing Theory and Applications, 2017.

1.1.5 Dynamic Selection in the indecision Regions

As stated above, Dynamic Selection techniques are sensitive to the definition of the region of competence, as well as the measures of competence. In a recent paper, Oliveira *et al.* (Oliveira *et al.*, 2017) raised the following issue in this context: DS techniques have difficulties evaluating the competence of classifiers when a test sample is located in an '*indecision region*', a region composed of samples from different classes. DS techniques may select classifiers with decision boundaries that do not cross the region of competence and thus, assign all the samples to one class which does not reflect the representation of the RoC. Oliveira et al. (Oliveira *et al.*, 2017) advances that: "*an ideal classifier would be the one that crosses the region of competence and correctly distinguish between the samples from the different classes.*"

Therefore, they designed a framework called "Frienemy Indecision Region Dynamic Ensemble Selection" for two-class problems (FIRE-DES). The method allows to detect if a test sample is located in an indecision region and, if so, prunes the pool of classifiers, pre-selecting classifiers with decision boundaries crossing the region of competence of the query sample (if such classifiers exist). After that, uses a DS technique from the set of pre-selected classifiers.

The next section discusses the usual ensemble generations methods used for the Dynamic Selection scheme. One method in particular is described in more detailed since a part of it is used in the construction of our system. A general discussion of the chapter is provided in section 1.4 before heading to the proposed system.

1.2 Ensemble Generation methods

1.2.1 The wisdom of crowds

“Can a collection of weak classifiers create a single strong one?" is a frequently asked question in Ensemble learning indeed. Surowiecki replies, that under certain controlled conditions, the aggregation of information from several sources, results in decisions that are often superior to those that could have been made by any single individual— (Rokach, 2010; Surowiecki *et al.*, 2007). According to Surowiecki, in order to be wise, the crowd should adhere to the following criteria (Rokach, 2010; Surowiecki *et al.*, 2007):

- **Diversity of opinion:** Each member should have private information even if it is just an eccentric interpretation of the known facts.
- **Independence:** Members’ opinions are not determined by the opinions of those around them.
- **Decentralization**: Members are able to specialize and draw conclusions based on local knowledge.
- **Aggregation**: Some mechanism exists for turning private judgments into a collective decision.

Ensemble methods have proven their worth through the years, the generation of base classifiers is the first step into building an ensemble. Therefore, this section explores the classical techniques and the newest techniques of generation methods, then comes a discussion part introducing our contribution in optimizing the pool generation.

1.2.2 Bagging

Proposed by Breiman (Breiman, 1996), Bagging is an acronym for 'Bootstrap AGGregatING'. It incorporates the advantages of Bootstrapping approaches (Efron & Tibshirani, 1993; Skurichina & Duin, 2002) and aggregating concepts by generating multiple versions of a classifier and using these versions to get an aggregated predictor(Breiman, 1996). The idea of the method is simple and builds *n* replicate training datasets by randomly sampling, with replacement, from the original training dataset. Since the sampling is conducted with replacement, some of the original instances appear more than once while some other original examples are not in the sample (Zhou, 2012). Because of such a property, some samples are similar because they are coming from the same original sample, but in the meantime, they are a bit different due to chance (Kaufman & Rousseeuw, 2009; Alpaydin, 2014). Thus, each replicated dataset is used to train one classifier member. The classifiers outputs are then combined via an appropriate fusion function. It is expected that 63.2% of the original training samples will be included in each replicate (Dos Santos *et al.*, 2008). Hence, the classifiers make different mistakes in the features space and then they are diverse.

Recently, (Walmsley *et al.*, 2018) proposed a version of Bagging modifying its bootstrapping process. It is in which the probability of an instance being selected during the re-sampling process is inversely proportional to its instance hardness (Smith *et al.*, 2014). The methods joins several data complexity measures and ensemble methods to improve the accuracy rate of the systems suffering from noisy data without sacrificing the samples located on the class boundaries .

1.2.3 Random Subspaces Method

The Random Subspaces Method (RSM) (Barandiaran, 1998) is considered to be a feature subset selection approach. It works by randomly choosing *N* different features from the training dataset obtaining N-dimensional random subspaces (Skurichina & Duin, 2002) from the original features space. Each random subspace is used to train one individual classifier. The *N* classifiers are usually combined by the majority voting rule. The advantages of using random subspace in the generation and combination of the classifier is appreciated when the number of training samples is small in comparison with the data dimensionality (Skurichina & Duin, 2002). The subspace dimensionality is smaller than the original features space while the number of training samples remains intact. It is useful when there are several redundant features, we may obtain better classifiers in random subspaces than in the original features space (Skurichina & Duin, 2002) which would be reflected on the quality of the classification in favor the random subspaces.

1.2.4 Boosting

There exists many variants of Boosting. We use AdaBoost (Adaptive Boosting) method in generating ensembles. Proposed by (Freund *et al.*, 1996), Adaboost is an iterative algorithm that combines classifiers having poor performance to get a better decision rule (Skurichina & Duin, 2002). The method assigns weights to each example contained in the training dataset and generates classifiers sequentially as opposed to Bagging which operates randomly and in a parallel way when sampling its training sets and constructing its classifiers. At each iteration, Boosting adjusts the weights of the miss-classified training samples by previous classifiers. Thus, the samples considered by previous classifiers as difficult for classification, will have higher chances to be put together, to form the training set for future classifiers (Freund *et al.*, 1996; Skurichina & Duin, 2002).

The final ensemble composed of all classifiers generated at each iteration is usually combined by majority voting or weighted voting (Dos Santos *et al.*, 2008; Skurichina & Duin, 2002; Freund *et al.*, 1996).

1.2.5 Oracle-based generation method

The consideration of the Oracle is a key issue in comparing between different techniques of dynamic selection. The Oracle is defined as an abstract function that always selects the classifier that predicts the correct label, for each instance, if such a classifier exists. In other words, it represents the ideal classifier selection scheme (Cruz, 2016). Its consideration is also present for the elaboration of several methods in MCS. However, several dynamic selection techniques produce a large difference of performances compared to the Oracle. This explains that, for a certain number of instances, the DS techniques are not able to select a competent or a set of competent classifiers despite the Oracles assurance of its presence in the pool. Therefore, *Souza et al.* tried to investigate in (Souza *et al.*, 2017) the reasons why the Oracle may not always be the best indicator in the search for a promising pool of classifiers for DS techniques.

Souza et al. proposed a new method of generating a pool called "Self-generating Hyperplanes (SGH)’ that guarantees an Oracle accuracy rate of 100% in the training set. It is an incremental ensemble generation method which generates binary classifiers by placing hyperplanes in the feature space until at least one classifier correctly classifies each training instance in the pool. This method is faster than classical ensemble methods, since the classifiers are not trained, and it can find the pool size automatically according to the training data (Souza *et al.*, 2017).

We provide its pseudo-code due to the fact that we used certain parts of the method that we modified in the creation of our classifiers, for instance the modified version will rely on lines 11, 12 and 13 from Algorithm 1.1.

The experiments of this work demonstrated that integrating Oracle information in the generation phase of an MCS has little impact on the gap between the accuracy rates of DS techniques and the Oracle (Souza *et al.*, 2017). Furthermore, for a theoretical limit of 100%, the DS techniques were only able to select a competent classifier for at most 85% of the instances, on average (Souza *et al.*, 2017). DCS techniques show struggles in choosing the most “competent” classifier, despite the existence for at least one for sure in the pool (trivially, the one that was created for that particular instance during the generation phase). The reason for this is that, the Oracle model relies on the global information confirming the existence of an adequate classifier for the task; whereas DCS techniques use only local data, as the local measures of competence and the K nearest neighbors to select the best classifier.

Algorithm 1.1 Self-generating Hyperplane Method (SGH), from (Souza *et al.*, 2017)

```
$\Gamma \leftarrow \{z_1, z_2, ..., z_N\}$ {Training dataset}
$C \leftarrow \{c_1, c_2, ..., c_{|C|}\}$ {Problem classes}
$Pool \leftarrow \{\}$
while $\Gamma \neq \{\}$ do
for $j \leftarrow 1, |C|$ do
    $R(j) \leftarrow centroid\,(c_j)$ {Centroid of class $j$}
end
$d \leftarrow max(pairwiseDistance(R))$ {Maximum distance between centroids}
$a, b \leftarrow findIndex(d)$
$midPoint \leftarrow (R(a) + R(b))/2$
$normal \leftarrow (R(a) - R(b))/d$
$w_p \leftarrow \{normal\}$ {perceptron p weights}
$\theta_p \leftarrow -midPoint \cdot normal$ {perceptron p bias}
$p \leftarrow perceptron(w_p, \theta_p)$
for $i \leftarrow 1, N$ do
    if $test(p, z_i) = label(z_i)$ then
        then {Perceptron p classifies instance i correctly}
        $\Gamma \leftarrow \Gamma - \{z_i\}$ {Excludes instance $i$ from dataset}
    end
end
$Pool \leftarrow Pool \cup \{p\}$ {Add Perceptron $p$ to the Pool}
end while
return $Pool$
```

Therefore, (Souza *et al.*, 2017) conclude that despite its use in the literature for such a task, the Oracle model is not the best guide in the search for a promising pool for DCS techniques, for the model is performed globally whilst DS techniques work with local data only (Souza *et al.*, 2017).

1.3 Summary, discussion and a brief introduction to the proposed system

The Oracle concept is an important element in the elaboration of Multiple Classifiers Systems. It contributes to both Ensemble generation and the classifiers selection. For DCS, it is considered as an upper theoretical limit for classification and usually determines the efficiency of the DCS techniques (Cruz *et al.*, 2018; Kuncheva, 2004b; Souza *et al.*, 2017). Moreover, the Oracle was used in the elaboration of several Dynamic Selection methods and was studied to investigate its characterization for DCS techniques (Souza *et al.*, 2017). Although its well reputation, it was found that its aspect of spotting the ideal classifier, operates on a global level as opposed to the local treatment of instances provided the DS techniques. Therefore, the Oracle was labeled as being not the best guide for generating a pool of classifiers for DCS (Souza *et al.*, 2017). This conclusion played an important role into bringing more motivational elements to our research question that aims to generate classifiers adapted to the context of dynamic selection focusing on the consideration of local information.

This being stated, the related work presented first, a general overview of the dynamic selection scheme by presenting the DS techniques and auxiliary information that aligns with the problematic of this thesis. Moreover, we introduced the different pool generation methods that exist and here we are narrowing down the research proposal.

The difference between the DS techniques reside in their definition of the Region of Competence, their selection criteria and the selection approach. The Selection criteria in the DS scheme is composed of two philosophies of considering the competence by different measures. In these individual-based measures of competence, the reliability of each base classifier is measured independently from the performance of the rest of the classifiers in the pool (Cruz *et al.*, 2018). These methods are full dependent on the methods that defined the region of competence such as the K-NN. Moreover, the distribution of *DSEL* has an important impact on the performance of the system. Group-based measures of competence on the other hand, focus on how the base classifier behaves along with the other classifiers in the pool (Cruz *et al.*, 2018) and therefore relate to the concept of relevance as opposed to the individual-based measures where they rely on the competence of the individual classifiers (Cruz *et al.*, 2018) .

Despite the variety of the DS techniques and the different components of this classification scheme, no algorithm is better than any other over all possible classes of problems (the "No Free Lunch" theorem (Corne & Knowles, 2003)). Thus, given the overview of the different research works conducted in the area of MCS, and the recent findings on the question of the locality of the classifiers; we believe we have satisfying elements and motivations to focus towards proposing a new pool of classifiers generation that is suitable for the Dynamic Selection Scheme. The proposed framework is provided in the next chapter with all the basics concepts that need to be known before a complete immersion into the method.

CHAPTER 2

TOWARDS LOCAL POOL GENERATION FOR DYNAMIC SELECTION

The present chapter exposes the proposed system. The first section introduces the concepts needed in the elaboration of the method to generate locally competent classifiers. The rest of the sections are divided between an overview of the method, the step by step description and illustrative examples of the different strategies used to answer the research question. A case study on a synthetic problem is also provided.

2.1 Basic concepts

To begin our discussion, we believe it is necessary to present the following illustrative basics concepts that we would refer to throughout the chapter.

2.1.1 Region of Competence in the context of this study

The Region of Competence for several DS techniques is defined using the $K-NN$, and strongly depends on the distribution of DSEL (Cruz *et al.*, 2018; Didaci & Giacinto, 2004). For our proposed system, the definition of the RoC is covered by the $K-NN$ and the value of $K = 7$ since it presented the best results for several DS techniques according to (Cruz *et al.*, 2011, 2018)

2.1.2 Indecision Region

As illustrated in Figure 2.1, a test sample is located in an *indecision region* when its region of competence is crossed by one or more classes boundaries, that is, when its region of competence has borderline samples of different classes (Oliveira *et al.*, 2017). Therefore, correctly classifying test samples located in *indecision regions* is a difficult task because most miss-classifications happen in areas near classes boundaries (Oliveira *et al.*, 2017). In fact, the classification performance of classifiers is strongly affected by the number of borderline samples (Oliveira *et al.*, 2017).

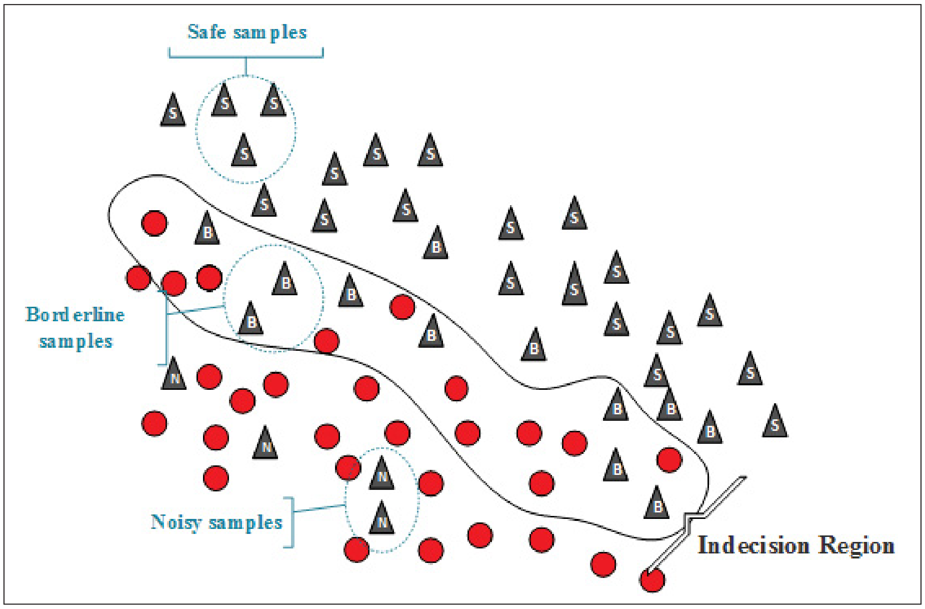


Figure 2.1 Three type of samples: safe samples (labeled as S), borderline samples (labeled as B), and noisy samples (labeled as N). The continuous line shows the indecision region(Adapted from (Oliveira *et al.*, 2017)).

2.1.3 frienemies samples

The definition of the frienemies concept is essential to the understanding of the proposed method. Indeed, according to (Oliveira *et al.*, 2017), two samples x_a and x_b are considered frienemies if : (1) they are located in the same region of competence of x_{query} (2) they are from a different class. Therefore, one of our research aims is to create classifiers that distinguish between them. Figure 2.2 is a representation of a region of competence with its different samples A, B, C, D, E, and F for the query x_{query}. The possible pairs of samples from different classes are (A, B), (A, D), (B, C), (B, E), (B, F), (C, D), (D, E), (D, F). They are called "frenemy samples" or "frienemies" given (Oliveira *et al.*, 2017)).

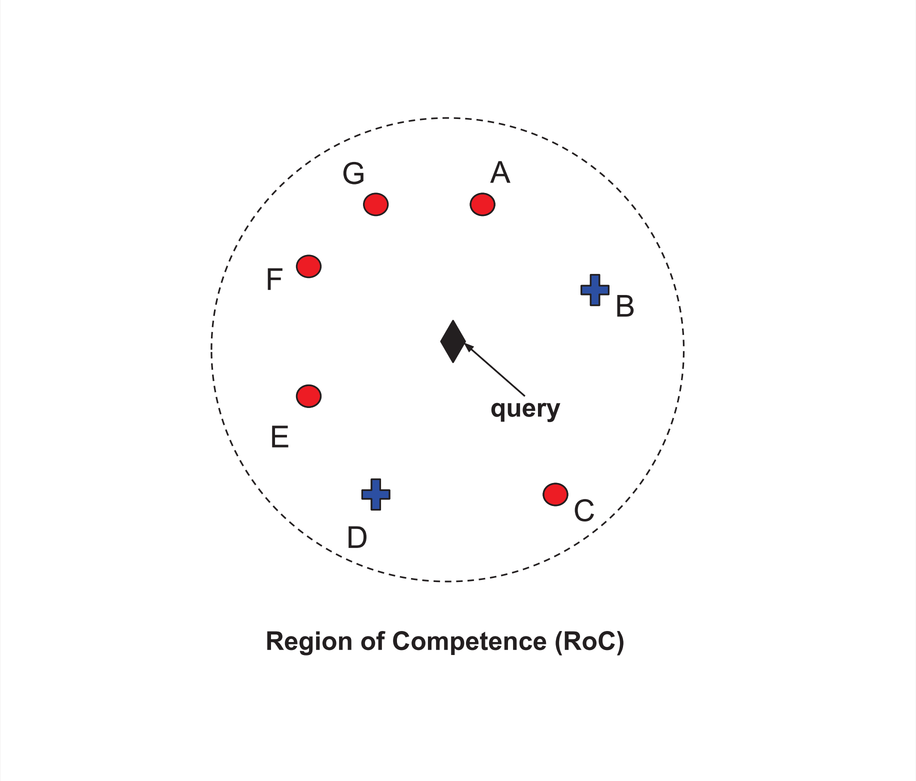


Figure 2.2 Representation of pairs of frienemies (A, B), (A, D), (B, C), (B, E), (B, F), (B,G) (C, D), (D, E), (D, F), (B,G) in the region of competence of the query in a black diamond shape (Adapted from (Oliveira *et al.*, 2017)).

2.1.4 Instance Hardness

Instance hardness is a fundamental concept in the elaboration of our proposed system. A study conducted by (Smith *el al*, 2014) states that the instances have a set of hardness properties that reflects the likelihood that they will be misclassified. Instance hardness measures on the other hand yield an indication on which of the samples in the datasets are hard to classify. In the same work, (Smith *et al.*, 2014) provides an instance level analysis of data complexity, giving us more insights about the different measures of instance hardness.

2.1.4.1 k Disagreeing Neighbors (kDN), an instance hardness measure

Let $T_e = \{x_1, x_2, ..., x_j\}$ be the test set composed of j elements. The strategy will operate as follows:

- Compute the instance hardness (IH) of each sample x_q using the kDisagreeing Neighbors (kDN) measure. The choice of this measure is justified by the highest correlation it presented with the probability that a given instance is miss-classified by different classification methods according to (Smith *et al.*, 2014; Cruz *et al.*, 2017). The kDN measure is the percentage of instances in an instance'neighborhood that do not share the same label as itself. Equation 3.1.5 shows the measure.

$$kDN(x_q) = \frac{|x_k : x_k \in KNN(x_q) \wedge t(x_k) \neq t(x_q)|}{K} \tag{2.1}$$

  where $KNN(x_q)$ is the set of K nearest neighbors of x_q, and x_k represents an instance in this neighborhood. $t(x_q)$ and $t(x_k)$ represent the target class of the instances x_q and x_k respectively. In our method, we consider in the beginning a neighborhood size $K = 7$ for the estimation of the kDN.

A basic example to illustrate the level of Hardness of the query (black diamond) in Figure 2.2 according to the $k-DN$ measure is $\frac{2}{7}$ if the query is red and $\frac{5}{7}$ if the query is blue, it represents the proportion of disagreement of the neighborhood with the label of the query.

2.2 The proposed Local Pool Generation for Dynamic Selection System

We propose in this section a way to create a pool of classifiers that is adapted to the scheme of Dynamic Selection. The pool generation method is then called: **D**ynamic **S**election **P**ool **G**eneration (**DSPG**). It is based on the following hypothesis:

- If we maximize the coverage of the features space in the indecision regions, considering local information, the classifiers generated within the training set would be able to high performances in generalization of **DS** techniques.

Therefore, we show in Figure 2.3 the general overview of the different stages of the method. The main steps are explained in details hereafter.

The proposed method (DSPG) generates the Dynamic Selection Local Pool **(DSLP)** that is adapted to the context of **DS**. Indeed, this method uses several strategies to guarantee the creation of locally competent classifiers. In our recent paper (Cruz *et al.*, 2017) (Appendix I), a deep analysis comparing the performances of Dynamic Selection and the plain $K-NN$ classifier was conducted where it was concluded that $K-NN$ performs better and faster than DS for samples with low instance hardness; whereas Dynamic Selection is more suitable for samples with a higher degree of instance hardness.

Accordingly, for the generalization phase, we designed a system that operates in two steps:

1. First, an Instance hardness analysis (IH) is conducted on the test samples based on the k Disagreeing Neighbors (kDN) measure. The samples that are located in homogeneous and safe regions ($IH = 0$) will be classified by KNN.
2. For the other ones, the Dynamic Selection scheme is used.

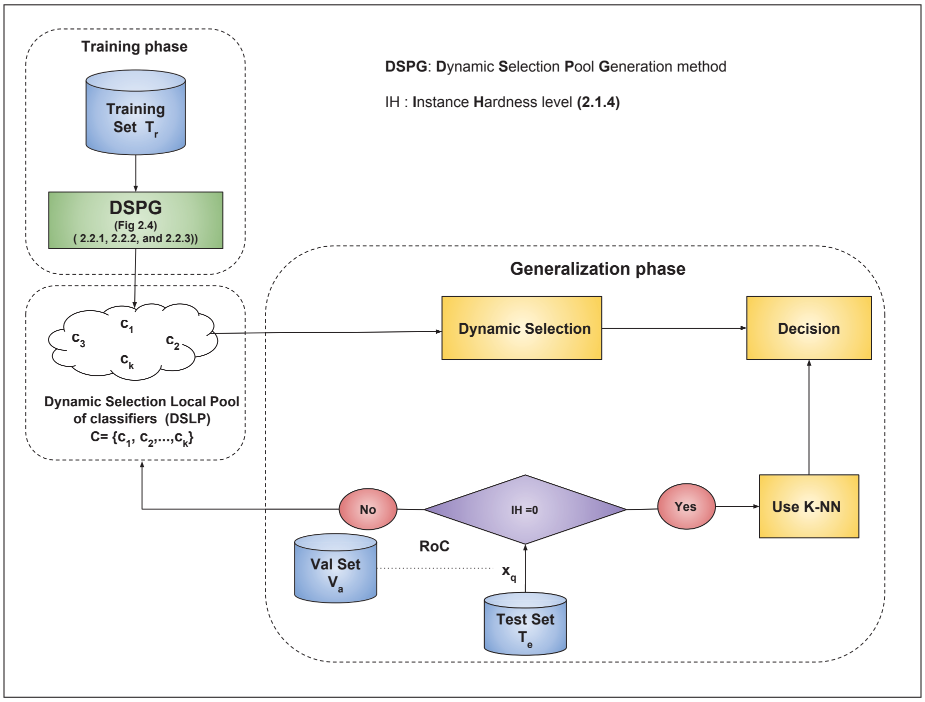

Training phase
DSPG: Dynamic Selection Pool Generation method
IH : Instance Hardness level (2.1.4)
Training Set T_r
DSPG
(Fig 2.4)
(2.2.1, 2.2.2, and 2.2.3))
Generalization phase
c_1
c_3
c_2
c_k
Dynamic Selection
Decision
Dynamic Selection Local Pool of classifiers (DSLP)
C= {c_1, c_2,...,c_k}
No
IH =0
Yes
Use K-NN
Val Set V_a
RoC
x_q
Test Set T_e

Figure 2.3 General overview of the proposed framework. In the training phase, we use the proposed Dynamic Selection Pool Generation method **(DSPG)** to create the Dynamic Selection Local Pool (**DSLP). ** Then, depending on the hardness of the test query sample in generalization, the system uses either the **K-NN** to take the decision or **DS** with the generated pool (**DSLP**) as suggested in [71].

It is worth noting from a general view, that the DSPG method showed in Figure 2.4 is conducted as follows: At first, from the training set T_r, separate easy samples from the hard ones according to the IH measure explained above. A sample is considered "easy" if the value of the Instance Hardness *(IH)* defined by the kDisagreeing Neighbors (kDN) measure is $IH = 0$ (i.e, located in a homogeneous region). Then, for every Hard sample x_j, we compute its RoC θ_j in V_a (used as $DSEL$) with a $K-NN$ rule.

The suggested method allows to create a Temporary Pool (TP) of classifiers dedicated to every specific hard sample, such that each classifier from TP crosses the RoC by linearly separating between the frienemies samples. Additionally, the method places the hyperplanes taking into consideration the proportion of samples belonging to each class.

Given the previous information regarding the frienemies separation and the creation of the temporary pool (TP), several questions regarding the pool generation arise:

Which pair of frienemies is the most suitable to the elaboration of a locally competent classifier(s)? How many classifiers should one generate? What are the criteria that are relevant to locally select the classifiers for a better generalization?

These questions, lead us to introduce a "local selection" mechanism in the context of dynamic selection. We conduct the local selection using different strategies to integrate the most adapted local classifier(s) from TP, and constitute the Dynamic Selection Local Pool (*DSLP*), a pool that covers the indecision regions in the features space.

The frienemies separation concept, the creation of the Temporary Pool as well as the local selection strategies are explained in details in subsections (2.2.2) and (2.2.3) respectively.

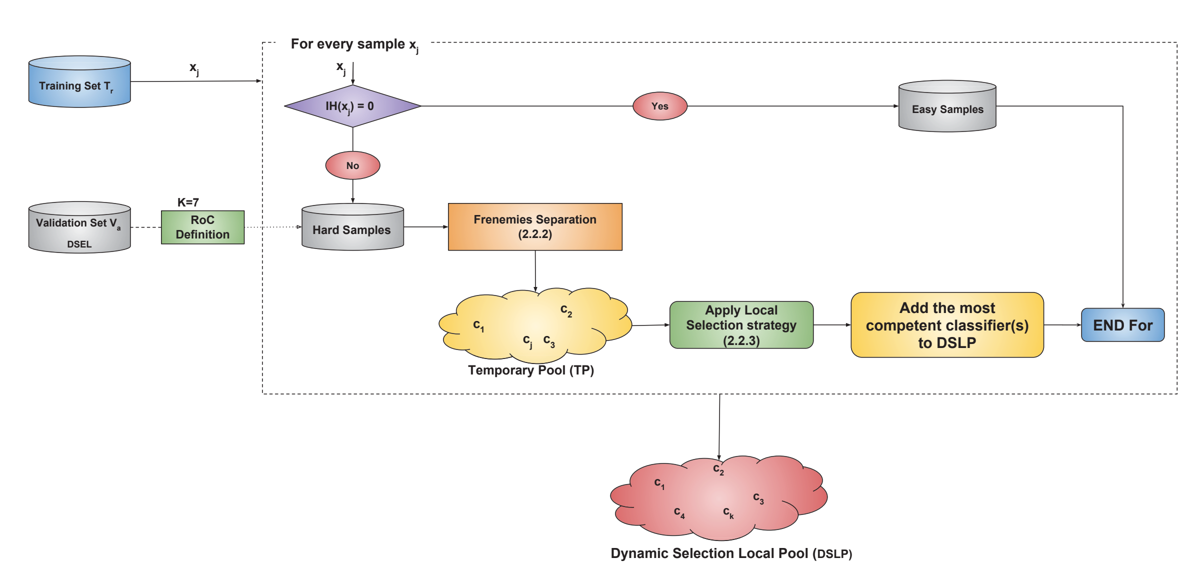


Figure 2.4 Summary of the Dynamic Selection Pool Generation (DSPG) part with the different selection strategies

2.2.1 How does the proposed local pool generation work?

In Algorithm 2.1 is presented the pseudo-code of the Dynamic Selection Pool Generation (**DSPG**) method that returns its Dynamic Selection Local Pool (**DSLP**), as explained in the previous diagrams. The step by step explanation of the algorithm is provided. The pairwise separation of the frienemies, the creation of the Temporary Pool (TP) and the different strategies of local selection are presented in separated sections.

Algorithm 2.1 Pseudo code of the Dynamic Selection Pool Generation technique

```
1  Input: $T_r; V_a$
2  Output: $DSLP$
3  $DSpool \leftarrow \{\}$
4  Separate easy samples from hard ones using the $kDN$ measure (2.3.1)
5  for each hard samples $x_j \in T_r$ do
6      $\theta_j \leftarrow$get the neighborhood $(x_j, V_a)$
7      Create a Temporary Pool $(TP)$ within $\theta_j$ , separating the frienemies according to
         (2.1)
8      Add the chosen classifier $c^*$ or the chosen ensemble of classifiers $C^*$ to $DSLP$
         according to the corresponding strategy explained in (2.2.3)
9  end
10 Return $(DSLP)$
```

In this procedure, we expect to generate a pool of classifiers that is based on local information. The strategy of the proposed method is given following the pseudo-code above. First, in line 1, we omit all the samples that are located in safe regions (homogeneous) according to the *kDN* measure of instance hardness explained in (Cruz *et al.*, 2017) and given in more details in 2.3.1 to separate between the easy samples and hard ones.

For each sample x_j from the Hard samples in the training set (line 2), we compute its Region of Competence (RoC) θ_j within the validation dataset (V_a, $DSEL$) using the K-Nearest neighbor method. In line 4, a Temporary Pool (**TP**) is created in the RoC; by doing a pairwise separation between the frienemies samples within RoC (the details of the pairwise separation are given in 2.2.1.

In line 5, we apply the local selection strategy on the temporary pool created earlier to find the most competent classifier(s) according to one of the proposed strategies we explained in (2.2.3.1), (2.2.3.2), (2.2.3.3), (2.2.3.4) and (2.2.3.5). The chosen classifier c^* or the chosen ensemble of classifiers C^* is (are) then added to the Pool that is suggested to be adapted to *DSLP*, the **D**ynamic **S**election **L**ocal **P**ool.

The proposed 5 strategies are based on 3 guides to perform the local selection, they are briefly described as follows and the details will be explored in subsection 2.2.3:

- **DCS as a guide for pool creation:** we use DCS techniques on the Temporary Pool within the RoC for x_j to find the most competent classifier c^* according to DCS. Then, depending whether it is **strategy 1** or **strategy 2** we choose to add c^* or not.
- **KNORA-E as a guide for pool creation:** in this case, representing our **strategy 3**, we rely on the decision of KNORA-E that acts as a local oracle regarding the ensemble of classifiers that will be added to the pool for each x_j.
- **Maximum number of well classified frienemies as a guide for pool creation:** for this guide, we expect the most competent classifier to be the one that distinguishes between a maximum pairs of frienemies and add the first **one** that meets this requirement to *DSLP* **(strategy 4)**. In **strategy 5**, **all** the classifiers that distinguish between a maximum number of frienemies are kept and added the pool.

In all cases, after adding the classifier(s) to *DSLP*, the algorithm keeps treating all the hard samples until its return the final Dynamic Selection Local Pool (line 7), that will be used in generalization (the generalization pseudo code is provided in 2.3).

2.2.2 A pairwise separation between frienemies

In this work, we suggest that the creation of local classifiers is conducted by the pairwise separation between the frienemies (Oliveira *et al.*, 2017) within the neighborhood θ_j (RoC) of a query x_j. Furthermore, we present a method to generate a set of hyperplanes that cross the RoC, inspired by the Oracle based generation method (Souza *et al.*, 2017).

The purpose of practicing a pairwise separation between the samples from the local region is motivated by the difficulties faced by the DS techniques selecting competent classifiers when they do not cross the region of competence; according to the observations of (Oliveira *et al.*, 2017).

Figure 2.5 shows the region of competence for a given query represented by a black diamond. Let X and Y be two points from class 1 (red circle) and class 2 (blue cross) respectively.

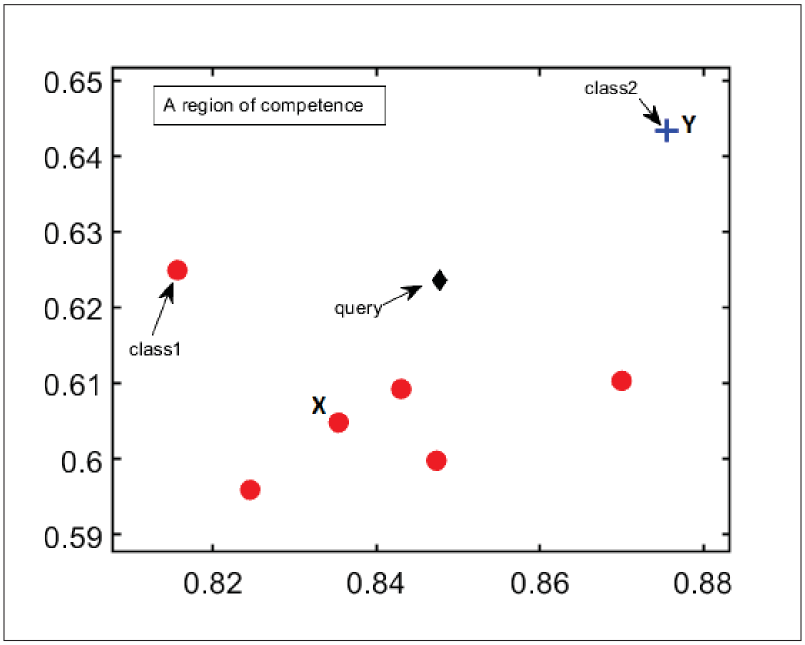


Figure 2.5 Region of competence for a given query represented by a black diamond. Red circle represents class 1, and blue cross represents class 2.

The separation of the frienemies X and Y, can be simply conducted by creating a hyperplane that crosses the segment $[X, Y]$. Knowing that the equation of the segment $[X, Y]$ is:

$$\lambda X + (1 - \lambda)Y \ \ , \lambda \in [0, 1] \tag{2.2}$$

The question that arises is: at which point should an hyperplane cross the segment $[X, Y]$, in order to obtain a proper separation? In other words, What could be the value of λ to have an adequate separation?

Note: According to the previous equation, for $\lambda = 1$, the hyperplane would cross the point X, and for $\lambda = 0$, the hyperplan would cross the point Y.

For this particular example, as there is only one sample from class 2, it would be more favorable for the hyperplane to cross the segment $[X, Y]$ at a point closer to Y, preferably at a distance that would be proportional to the ratio of sample of class 2 which is a minority over the total number of neighbors K. This can be accomplished by simply setting $\lambda = \frac{|minority|}{K}$ $(minority = |class\,2|)$ as $|class\,2|$ is the number of samples represented in blue cross which symbolizes the cardinality of the minority samples. This lead us to grant to λ the value that represents the proportion of disagreement between the samples from different classes within the neighborhood.

Thus, the hyperplane generated in Figure 2.4 crosses $[X,Y]$ at a point closer to Y for a value of $\lambda = \frac{1}{7}$.

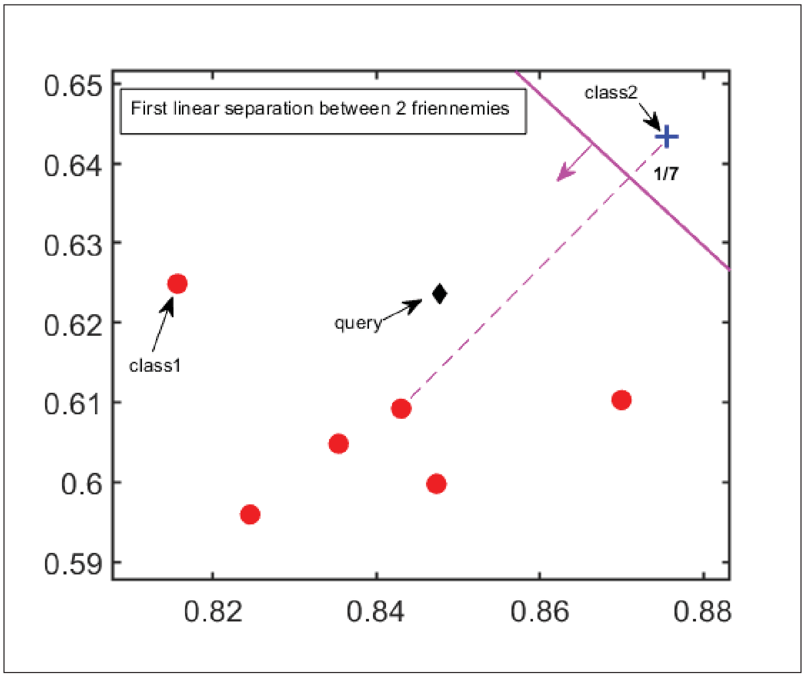


Figure 2.6 The frienemies separation proposed that is closer to the minority class (blue cross) with the pink classifier for a $\lambda = \frac{1}{7}$.

To define the perceptron's weights without explicitly training them, we have used a similar heuristic as the one proposed by (Souza *et al.*, 2017). However, we use a different way to calculate the bias; taking into account the proportion of the disagreement between the frenemy samples in the Region of Competence. In fact, the heuristic proposed by (Souza *et al.*, 2017) considers the scalar product between the midpoint and the normalized distance vector between the centroids in the bias calculation. In our case, we modified the position of the hyperplane according to the proportions of the samples of different classes in the RoC, taking into consideration the minority class. A visual illustration is provided in Figure 2.7 challenging the consideration of the midpoint in such cases.

For all the pairs of frienemies X and Y (X of class (1) and Y of class (2)) from the neighborhood θ_q, we calculate the weight w_j and bias μ_j as follows according to the equations (2.3) and (2.4):

$$w_j = \frac{X - Y}{||X - Y||} \tag{2.3}$$

$$\mu_j = -w_j.(\lambda X + (1 - \lambda)Y) \tag{2.4}$$

with $\lambda = \frac{|minority|}{K}$.

Observation

Regarding the value of λ, a trivial value is the midpoint ($\lambda = 0.5$). Through Figure 2.7, we observe that the consideration of the midpoint as conducted by (Souza *et al.*, 2017) is not well representative to the reality of the region of competence, contrary to what have been proposed. We can see that the gray classifier gets a mistake in classifying one member of the neighborhood, as opposed to the blue one.

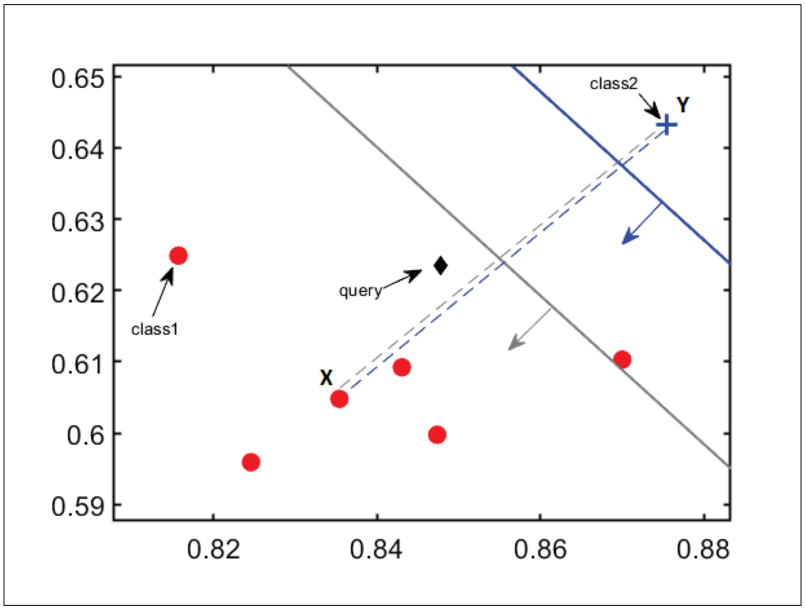


Figure 2.7 The difference between the frienemies separation in the midpoint (in gray) and the proposed separation (blue cross)

2.2.3 Strategies for local selection of classifiers

In this part, we introduce five new ways that aim to locally select one classifier (or more) from the Temporary Pool (TP) mentioned in the previous subsection. To do so, we created different guides or criteria to follow, 3 of them are based on the Dynamic Selection scheme and the rest rely on the frienemies concept as summarized in Table 2.1.

Table 2.1 A summary of the strategies used as guides to locally select the classifier(s) from the Temporary Pool TP and construct DSLP

Strategy	**Guide**	**Local Classifier's Selection Scheme**
Strategy 1	DCS	Add the most competent classifier from TP to keep, according to DCS measure of competence, only if it classifies well the sample
Strategy 2	DCS	Add the most competent classifier from TP to keep, according to DCS measure of competence
Strategy 3	KNORA-E	Let KNORA-E decide which subset of classifiers from TP to keep
Strategy 4	frienemies	Consider one of the classifiers from TP that classifies correctly the maximum number of frienemies
Strategy 5	frienemies	Consider all the classifers from TP that classify correctly the maximum number of frienemies

Indeed, we count five local selection strategies motivated by three different guides : DCS techniques, KNORA-E and the frienemies concept. The detailed explanations of the five strategies, their motivations, their advantages and disadvantages are given in the next subsections.

2.2.3.1 Strategy 1: DCS as a guide for local selection, no errors allowed

In this first strategy, we use the DCS measures of competence themselves as guides, in order to find the most competent classifier within the RoC for each hard sample.

The reason behind this choice is to **mimic** the behavior of the DCS technique given the local region. In fact, it was an intuitive direction to follow, to replicate the mechanisms of DCS within the temporary pool, so that it‘s assured not only to have a classifier that passes the region of competence, but also a classifier that would be competent according to each DCS competence rule.

This strategy comes with a local selection mechanism as shown in Figure 2.8. In fact, after creating the Temporary Pool for a certain hard sample in RoC from the training set, we apply DCS to find the most competent classifier locally c^*. However, c^* is added to the Dynamic Selection Local Pool (*DSLP*) only if it classifies well x_j , if it doesn’t then, the method moves to another sample to treat until no hard samples are left.

The advantage of the first local selection mechanism is that : (1) all the classifiers in the pool cross the region of competence, (2) all the classifiers are defined as most competent by the DCS technique and (3) all the classifiers classified well the query in training. However, as a disadvantage, some regions in the features space may not be covered, for the cases where the c^* is not added to *DSLP* due to the constraint of well classifying the sample.

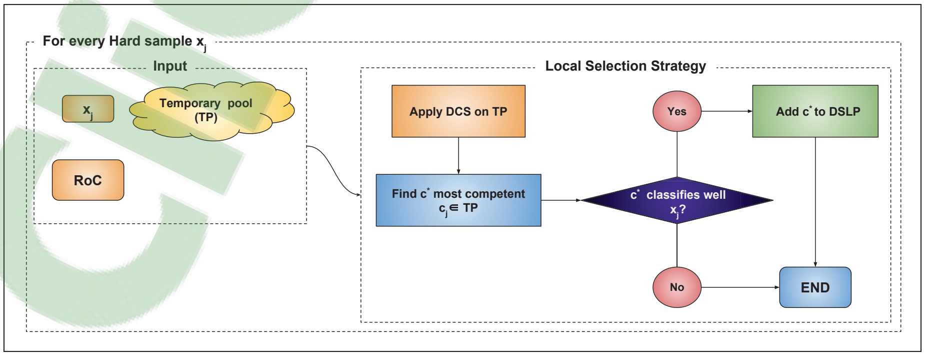


Figure 2.8 Local Selection Strategy 1

2.2.3.2 Strategy 2 : DCS as a guide for local selection

The second strategy is similar to the first one. It complies to the DCS scheme by applying the DCS techniques applied with the Temporary Pool to find the most competent classifier c^* within RoC for every hard sample. However, its local selection mechanism differers from the first strategy by keeping c^* whether it classifies well x_j or it doesn't.

On the other hand, the second selection of this strategy provides the following advantages: (1) all the classifiers in the pool cross the region of competence, (2) all the classifiers are defined as most competent by the DCS technique and (3) in all times, the features space is covered by these locally competent classifiers. However, due to the hardness of the samples, the geometrical properties of the problem and the quality of the samples located in the Region of Competence, some of the classifiers that are selected locally may fail in generalization.

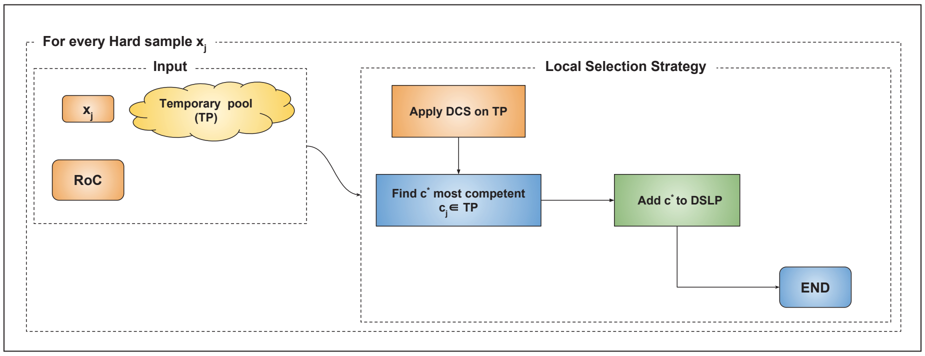


Figure 2.9 Local Selection Strategy 2

2.2.3.3 Strategy 3: KNORA-E as a guide for local selection

In this third strategy, we consider KNORA-E as a guide in the selection of the locally competent classifiers. We chose KNORA-E in particular because it is known as one of the top Dynamic Ensemble selection techniques according to (Cruz *et al.*, 2017, 2018; Oliveira *et al.*, 2017). Moreover, its mechanism is based on the application of the Oracle property when selecting the K-Nearest Oracles ensemble, composed of classifiers that correctly classify a given sample from the Region of Competence (Cruz *et al.*, 2017).

This selection mechanism, in the context of local pool generation works as follows: after passing by the same process of separating easy from hard samples according to the instance hardness measure, and generating the Temporary Pool (TP) for x_j within the RoC; the K-Nearest Oracles Eliminate (KNORA-E) is used to select a subset of competent classifiers from TP to be added to the Dynamic Selection Local Pool as shown in Figure 2.9.

One of the main advantages of using KNORA-E is to have several classifiers that correctly classify all the samples within the region of competence, as well as exploiting the concept of local Oracle. However, since the neighborhood could be reduced if there is no classifier that performs perfectly within the neighborhood according to KNORA-E properties; the reliability of the decision regarding the most competent ensemble within the neighborhood would not always be representative.

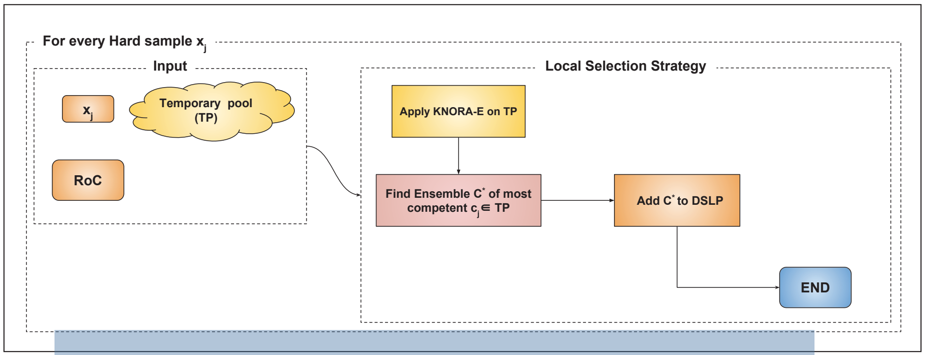


Figure 2.10 Local Selection Strategy 3

2.2.3.4 Strategy 4 : frienemies distinction as a guide for local selection, one classifier allowed

In this strategy, we test the ability of the classifiers to separate the maximum number of frenemy samples. The first part of the method remains the same as the previous ones;it means that we separate hard from easy samples according to their belonging to safe regions. The only difference between this proposed strategy and the others, resides in the criteria of selection to construct the pool of classifiers.

We evaluate the classifiers from TP according to how many frienemies they could correctly classify. The first classifier that scores the maximum number of well classified frienemies, is selected to be added to the Dynamic Selection Local Pool (*DSLP*).

The main idea behind this strategy is motivated by the FIRE-DES framework conducted by (Oliveira *et al.*, 2017), where in their online pruning, they kept temporarily classifiers that could distinguish between at least one pair of frienemies to classify the specific test sample. In our proposition, we used this property for locally creating these classifiers with an upgraded feature: the classifiers kept should distinguish between the pairs of frienemies. Once this condition is applied, one classifier is kept (strategy 4) or all those which satisfy this condition will be part of *DSLP* (strategy 5).

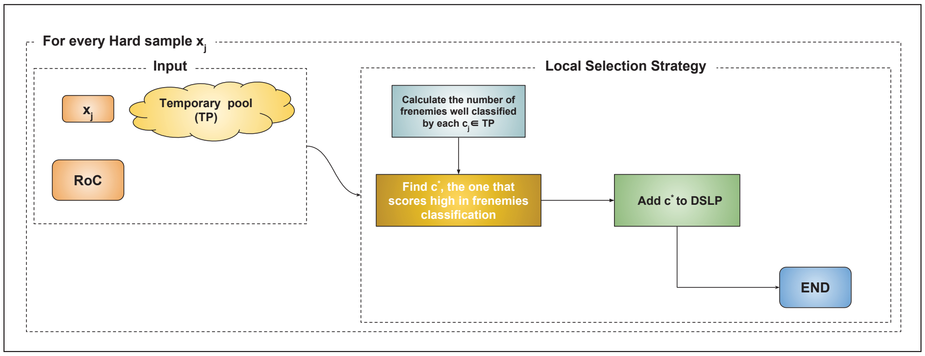

For every Hard sample x_j
Input
x_j
Temporary pool (TP)
RoC
Local Selection Strategy
Calculate the number of frenemies well classified by each $c_j \in$ TP
Find c^*, the one that scores high in frenemies classification
Add c^* to DSLP
END

Figure 2.11 Local Selection Strategy 4

2.2.3.5 Strategy 5 : frienemies distinction as a guide for local selection, multiple classifiers allowed

For this part, it works similarly as the previous strategy, the only difference is that we enlarge the spectrum by keeping all the classifiers that achieve a maximum score in the distinction between the frienemies. Thus, we would have more than one classifier that perfectly provides a local linear separation between the classes in the local region of competence.

Given the motivations cited above, in a recent paper (*Cruz et al., 2018*) stated that "*Ideally, a local competent classifier would be able to distinguish between all the frienemies pairs in the region of competence. Thus, being able to separate between the two classes locally* ". In this case, we aimed to maximize the presence of "ideal classifiers" as stated before.

Given the description of this strategy, its motivations and advantages, it is worth mentioning that one of its possible drawbacks (selecting all the classifiers that perfectly distinguish between all the frienemies in RoC) could lead to a DSLP of high cardinality, since the first purpose of the pairwise frienemies separation detailed in 2.2.1 is to enforce the property of maximizing the distinction of frienemies in an indecision region of competence.

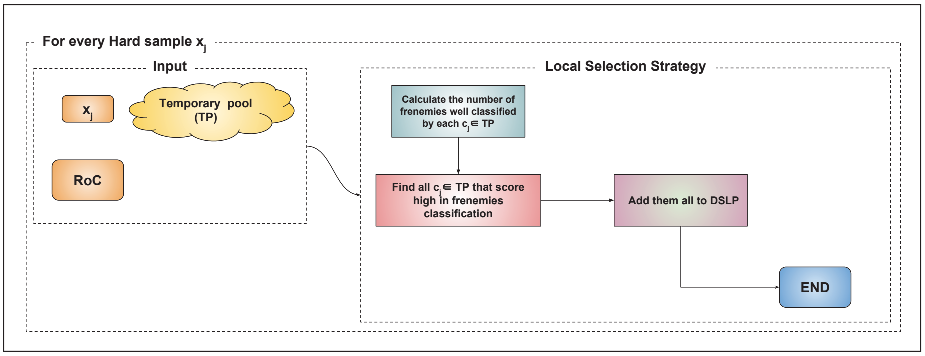

For every Hard sample x_j
Input
x_j
Temporary pool (TP)
RoC
Local Selection Strategy
Calculate the number of frenemies well classified by each $c_j \in$ TP
Find all $c_j \in$ TP that score high in frenemies classification
Add them all to DSLP
END

Figure 2.12 Local Selection Strategy 5

2.3 The generalization phase

After the pool generation, comes the part where we test its performance in generalization. As explained in the beginning of this section, we choose to exploit the instance hardness metric to define which samples will be classified by the KNN and which ones are going to be treated by the DS technique. For an instance which its hardness level equals to $IH = 0$ will be treated by KNN and the other samples belonging to indecision region will be assigned to the Dynamic Selection and are treated by the previously generated pool of classifiers. Algorithm 2.2 presents the pseudo code of this step. V_a represents the Validation dataset that we use as $DSEL$.

Algorithm 2.2 The joint use of the K-NN rule and DS techniques in generalization depending on the instance hardness level

```
**Input:** $x_{query}$,$V_a$, $DSLP$
$\theta \leftarrow$ Compute the Region of Competence of $x_{query}$ in $V_a$
**if** $IH(x_{query} = 0)$ **then**
    Apply the $K-NN$ rule
    **else**
    Apply $DS$ technique over $DSLP$
**end**
**Return** label($x_{query}$)
```

2.4 Case study:The P2 problem

This section includes a case study on a well known synthetic dataset named the "P2 problem". P2 is a two-class problem, presented by Valentini (Cruz *et al.*, 2015b; Valentini, 2005), in which each class is defined in multiple decision regions delimited by polynomial and trigonometric functions (Equation 6.1). As in (Cruz *et al.*, 2015b; Henniges *et al.*, 2005), E4 was modified so that the area of each class was approximately equal.

The P2 problem is illustrated in Figure 2.13 with the decision boundaries. We acknowledge that it is impossible to solve this problem using linear classifiers(Cruz *et al.*, 2015b; Valentini, 2005). The performance of the best possible linear classifier is around 50% (static selection)

(Cruz *et al.*, 2015b). In this explanatory example in Figure 2.13 , the P2 problem is generated as follows: 750 samples for training, 1250 for validation and 500 samples for testing.

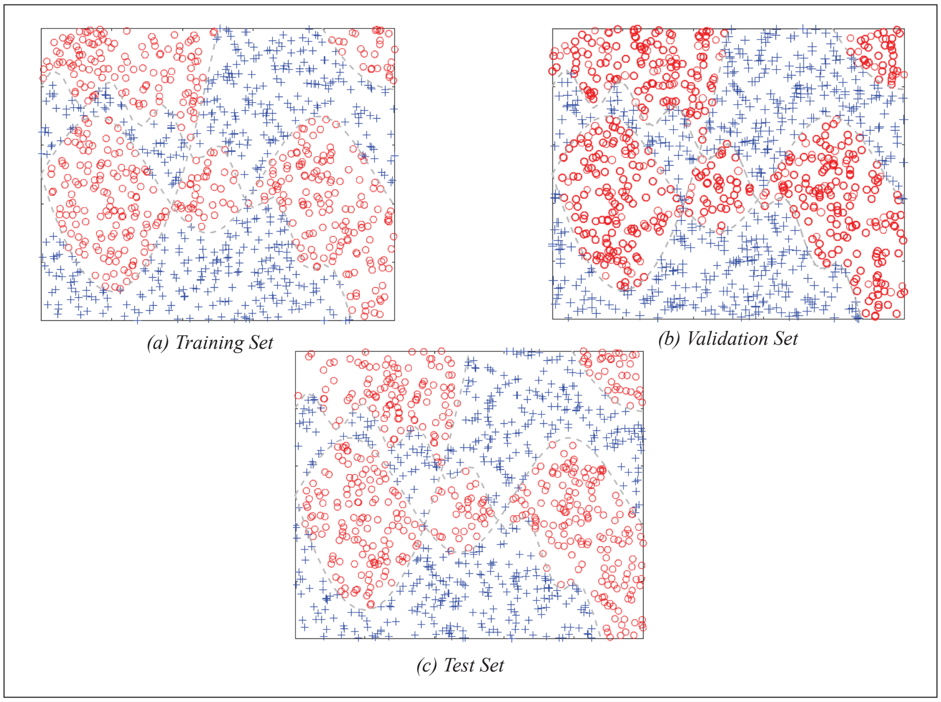


Figure 2.13 The P2 Problem with decision boundaries, the red circle refers to class 1

$$E1(x) = sin(x) + 5 \tag{2.5}$$

$$E2(x) = (x-2)^2 + 1 \tag{2.6}$$

$$E3(x) = -0.1x^2 + 0.6\,sin\,(4x) + 8 \tag{2.7}$$

$$E4(x) = \frac{(x-10)^2}{2} + 7.902 \tag{2.8}$$

2.4.1 Local Pool Generation for P2

The purpose of this work consists on generating locally competent classifiers that comply to the Dynamic selection scheme. It is worth reminding that the competence in this contexts means that the generated classifiers have to pass and cross the region of competence; whereas the compliance to the Dynamic Selection scheme means that we take local information and the properties of the DS techniques to have the most suited classifier or ensemble of classifiers for each region of competence. For a better understanding of the method, Figure 2.14 shows an example of visual steps of the first **DSPG** method, for the synthetic P2 problem.

For this illustration, we showed the first strategy (DCS as a guide for local selection) with LCA in wrapper. For generalization, we jointly used KNN classifier and LCA given the hardness of the queries. The detailed explanation will follow.

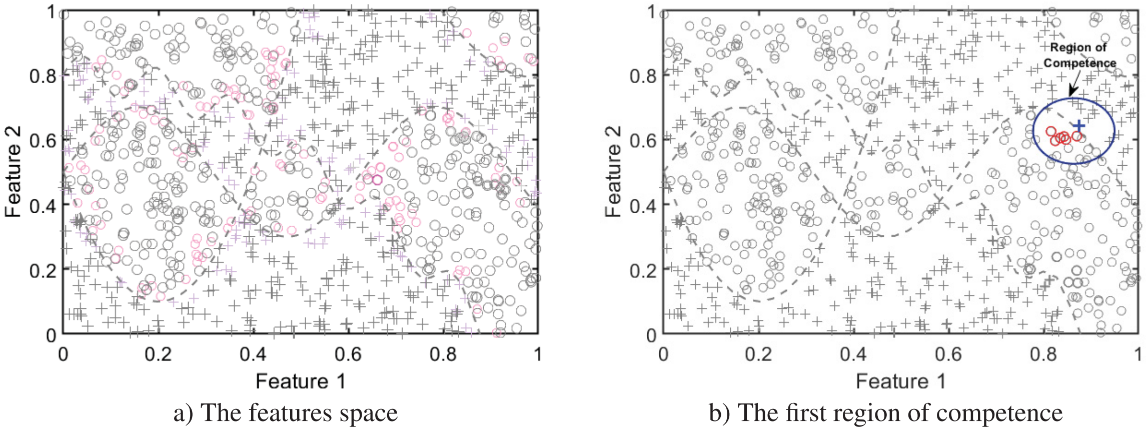
At this stage, we assume that the separation between the easy and hard samples has already been conducted. Figure 2.14 (a) reflects the features spaces, the gray samples represent DSEL, the light pink circles and purple crosses represent our hard instances from class 1 and class 2 respectively. We can clearly see that they are located on the decision boundaries which means that, they are located in indecision regions, according to (*Oliveira et al., 2017*).

In the first iteration:

The first step is to select a query to be treated, and its region of competence from *DSEL*, as illustrated in figure 2.14(b).

For this query, we apply the frienemies separation method to create the Temporary Pool (TP) (Figure 2.15(a)). The region of competence is encircled in blue and the classifiers generated are in the same color. We can see that there are several potential competent classifiers that distinguish perfectly between the samples of different classes. It optimistically means that, all the components are ready for the DS technique to choose the best classifier within the neighborhood.

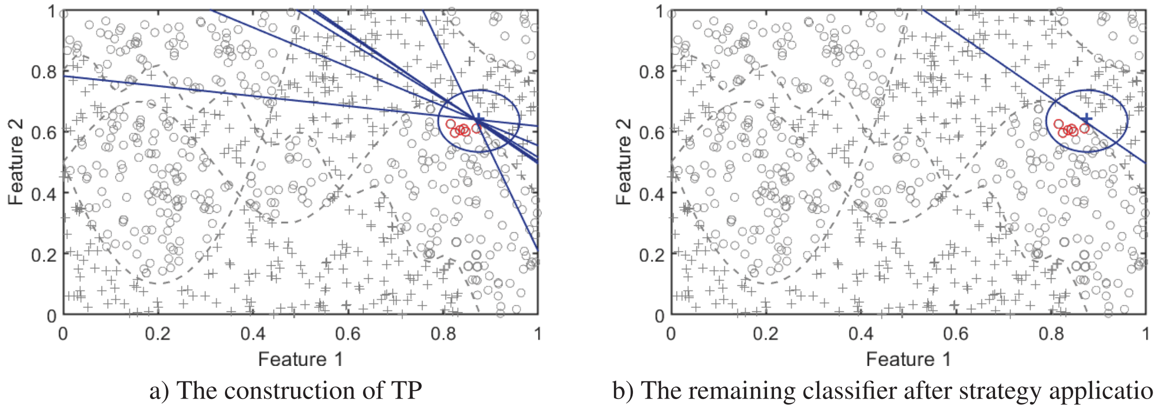
We then apply the DS technique (LCA in our case) on TP according to strategy 1: DCS as a guide for local selection of classifiers, no errors allowed. Only the selected classifier by LCA will remain, as shown in Figure 2.15b), and is therefore, added to the Dynamic Selection Local Pool (DSLP).



a) The features space

b) The first region of competence

Figure 2.14 A Region of Competence in an indecision region from the P2 problem



a) The construction of TP

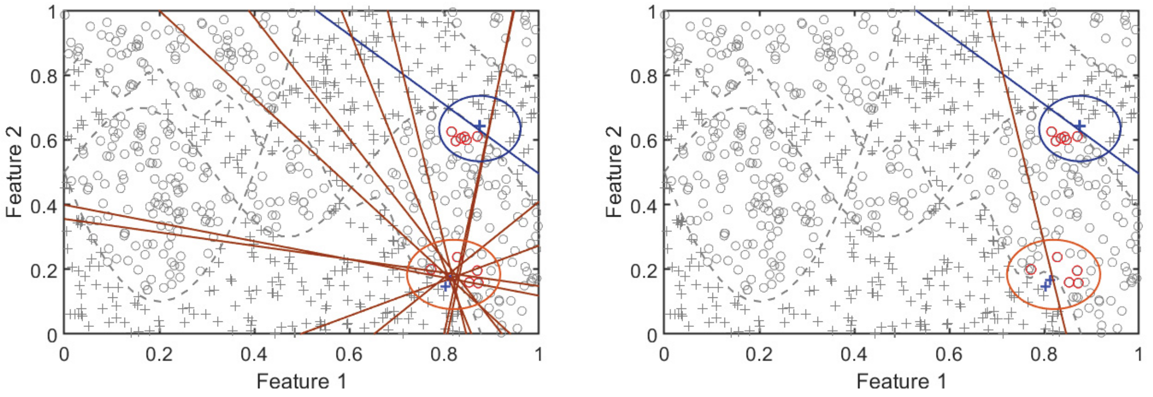
b) The remaining classifier after strategy application

Figure 2.15 First iteration: the generation of a Temporary Pool (TP) for a hard sample in a Region of Competence and the local selection of the most competent classifier to be added to DSLP

In the second iteration:

In the same way then the first iteration, another query is selected, and we apply the frienemies separation method to obtain the corresponding TP, as shown in Figure 2.16(a) (in brown).

After the application of DCS, only one classifier is added to DSLP as we can see in Figure 2.16(b). Again, the colors are meant to make the distinction between the regions of competence and the classifiers generated for each specific query x_q which are of the same color.



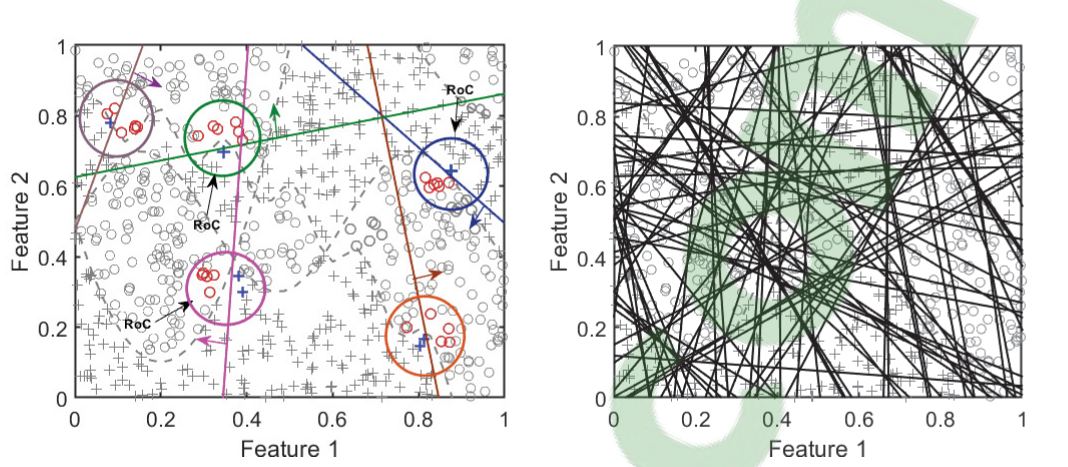
a) Iteration 2: TP construction

b) Iteration 2: remaining classifiers

Figure 2.16 The second iteration: the generation of a Temporary Pool (TP) for the second hard sample in a Region of Competence followed by the chosen classifier to integrate DSLP

In the next iterations:

Figure 2.17(a) represents the subset of classifiers obtained after the first five (5) iterations. Once all the hard samples are treated, we can see the final output in 2.17(b), as the feature space is well covered for this specific scenario. The features space has been covered with exactly **84** locally competent classifiers. The accuracy rate in generalization for the following replication was **94%**. For the same replication, the accuracy rate using Bagging was **89.1%** and **92.2%** for KNN (K=7).



a) The chosen classifiers within their RoCs

b) The final DSLP

Figure 2.17 A snapshot of the classifiers chosen applying the first local selection strategy and a final coverage of the space (b) of that specific replication

2.4.2 Case study summary

This case study falls in the context of enhancing the capacity of ensembles to learn locally, given a region of competence. It was motivated by the nonexistence of ensemble generation methods that are meant for Dynamic Selection. Previous works (Cruz *et al.*, 2018; Oliveira *et al.*, 2017; Cruz *et al.*, 2017) have underlined this gap and we aimed to take into consideration several of the previous works recommendations in order to create an informed classification system that takes into consideration the composition of the region of competences to generate classifiers that are contained within the indecision areas.

We provided an illustrative explanation to the frienemies pairwise separation given the region of competence for a given hard sample x_q. We also showed the final coverage of the features space when using the LCA properties within the pool generation, chosen as a guide for local selection of classifiers.

We can expect that separating the frienemies samples taking into account the minority class, maximizes the abilities of the classifiers to make the distinction between the samples. It also aligns with the DS technique that use the neighborhood for decision making. However, the distribution of *DSEL* and the geometrical configuration of the problem have a high impact on the definition of the neighborhood and therefore, affect directly the competence of the classifiers.

This case study explained step by step the evolution of the proposed method for the first strategy on the synthetic problem P2. The next chapter will present the experimental protocol, the different datasets used and the results of the strategies proposed. We also provide a comparative study between the results of the local selection strategies proposed and the results of state of the art pool generation methods.

2.5 Discussion

The proposed method consists in generating a pool of local classifiers, by taking into account local information. We expect the proposed system to have the following advantages:

1. **The presence of local competent classifiers;** the locality is justified by the generation of hyperplanes within the Region of Competence (RoC) of each query sample x_j, that is treated. The classifier(s) that has (have) the highest performance by any of the previous strategies is (are) considered as c^* or C^*, and is (are) included to the the Dynamic Selection Local Pool (**DSLP**).
2. The proposed techniques take into consideration several guides for local selection presented by the **DCS technique**s (strategy 1 and 2), **KNORA-E** (strategy 3) or the **frienemies distinction** (strategy 4 and 5).
3. The method is iterative and needs **no training**, since the parameters of the perceptrons are determined by the proposed heuristic inspired by the Oracle based ensemble generation method in (Souza *et al.*, 2017) with a modification of the position of the hyperplane, as we take into consideration the proportion of samples of different classes within the local region.
4. The proposed method in this work provides a **local coverage of the features space**, and focuses on creating hyperplanes for indecision regions; knowing that the safe regions (homogeneous) will be treated in generalization directly by the $K-NN$. This falls in the recommendation presented by (Cruz *el al.*, 2017) regarding the joint use of $K-NN$ and DS, given the hardness of the samples.

These are the direct advantages of generating locally competent classifiers; the experimental results will provide us insights about the behavior of **DSLP**in generalization compared to the baseline techniques. Moreover, we will see if the DS techniques are able to identify these locally competent classifiers and use them for hard instances predictions.

CHAPTER 3

EXPERIMENTS AND RESULTS

3.1 Experimental protocol

In the pursuit of elaborating a pool generation method based on local information in the context of Dynamic Selection, the study was performed using a test bed composed of 17, 2-class problems public datasets. Ten of them are from the UCI machine learning repository (Bache & Lichman, 2013), two from the Ludmila Kuncheva Collection (LCK) (Kuncheva, 2004a) of real medical data, two from the STATLOG project (KING *et al.*, 1995), one from PRTOOLS and two from the Knowledge Extraction based on Evolutionary Learning (KEEL) repository (Alcalá-Fdez *et al.*, 2011). The key features of the datasets are presented in Table 3.1. 20 replications were conducted on each dataset. For each replication, the datasets were randomly divided as follows: 50% for training, 25% for the dynamic selection dataset (DSEL) and 25% for the test set for the local pool generation method. The mention (IR) represents the Imbalance ratio for each dataset.

The results were obtained using seven state-of-the-art DCS methods for all the strategies presented in 2.2.3: Overall Local Accuracy (OLA) (Woods *et al.*, 1997), Local Class Accuracy (LCA)(Woods *et al.*, 1997), Multiple Classifier Behavior (MCB) (Huang & Suen, 1995), Apriori (Giacinto & Roli, 1999), Aposteriori (Giacinto & Roli, 1999), Modified Local Accuracy (MLA) (Smits, 2002) and DCS-Rank (Woods *et al.*, 1997; Smits, 2002). We also used the following DES techniques: The K-Nearest Oracles Eliminate (KNORA-E) (Ko *et al.*, 2008) and The K-Nearest Oracles Union (KNORA-U) (Ko *et al.*, 2008). The neighborhood size *K* for each of the *DCS* and *DES* techniques is fixed to 7, since they performed the best with this setting as reported according to the survey (Cruz *et al.*, 2015a, 2018; Cruz, 2016).

Table 3.1 Key features of the 17 datasets used for the experiments, IR represents the Imabalance Ratio.

Database	No. of Instances	Dimensionality	Source	IR
Adult	48842	14	UCI	1.25
Blood transfusion	748	4	UCI	3.17
Breast (WDBC)	568	30	UCI	1.67
German credit	1000	20	STATLOG	2.33
Haberman's Survival	306	3	UCI	2.8
Heart	270	13	STATLOG	1.26
ILPD	583	10	UCI	2.48
Ionosphere	315	34	UCI	1.75
Laryngeal1	213	16	LKC	1.65
Lithuanian	1000	2	PRTOOLS	1
Liver Disorders	345	6	UCI	1.39
Mammographic	961	5	KEEL	1.06
Monk2	4322	6	KEEL	1.11
Pima	768	8	UCI	1.87
Sonar	208	60	UCI	1.16
Vertebral Column	310	6	UCI	2.12
Weaning	302	17	LKC	1

To compare the different solutions provided for a defined problem, we need to properly define a quantitative way to evaluate the systems. We will base our comparison on Demšar's paper on Statistical Comparisons of Classifiers over Multiple Data Sets (Demšar, 2006) using the sign rank test, and the Bonferoni-Dunn post-hoc treatment, as well as wins, ties and losses comparisons.

The proposed system depends on the following parameters: The dynamic selection method, and the region of competence of the size (K). Moreover, it was given the same value for the purpose of later comparison.

3.2 Results analyses and discussions

In this section, we study the results of the combination of the $K-NN$ classifier and the dynamic selection algorithms under the different pool generations strategies. Indeed, following the recommendations of (Cruz *et al.*, 2017), we created a system that either uses the $K-NN$ classifier or DS, according to the level of hardness of the samples. The instance hardness is defined in the experiments as $IH = 0$ (safe region). The reason behind this strategy is to push the system to use $K-NN$ in homogeneous regions and therefore accelerate the processing time.

Figure 3.1 illustrates the range of the IH measure into deciding whether we use $K-NN$ or DS techniques, for this experiments we decide to use $K-NN$ over DS only in safe regions $(IH = 0)$.

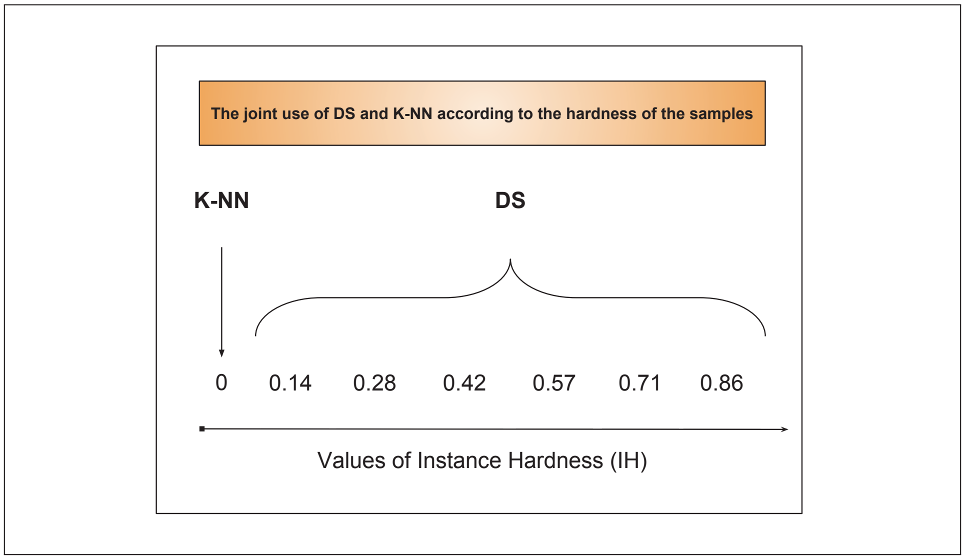

The joint use of DS and K-NN according to the hardness of the samples
K-NN
DS
0
0.14
0.28
0.42
0.57
0.71
0.86
Values of Instance Hardness (IH)

Figure 3.1 Instance Hardness measure for the joint use of $K-NN$ and DS in generalization.

Our research question concerns the local pool generation of classifiers for Dynamic Selection. We proposed a system composed of several phases in order to solve the problem. We take into consideration local information within the Region of competence of each sample located in an indecision region for two class problems to:

- Create a Temporary Pool (TP) to separate between the frienemies in a pairwise scheme. For a neighborhood size equals to 7, the number of local temporary classifiers generated is either 6, 10 or 12 classifiers corresponding to the number of samples from each class (1,6), (2,5) and (3,4) respectively.
- Adopt 5 local selection strategies in order to choose either one classifier or an ensemble of classifiers that are competent within TP.

3.2.1 Comparison between the proposed local pool generation strategies and the state of the art generation methods

In this part, we conduct different statistical tests in order to have a wide understanding of the results provided by the experiments.

As a robust multi-comparison global approach, the Friedman test was conducted as well as the Bonferoni-Dunn post hoc test (Demšar, 2006) to illustrate the differences between the strategies and the two state of the art pool generation methods (Bagging and SGH) for DS techniques. The results of this test are shown in Figure 3.2.

The aim is to find which of the proposed strategies suits best the Local Pool Generation and which is comparable to the state of the art methods. The baseline pool generation methods are Bagging in which we generate 100 perceptrons and SGH where it generates in average less than 5 classifiers whereas in our proposed method, the number of classifiers is proportional to the number of hard samples (the details of the number of classifiers is in Appendix II). This statistical test helps to find the strategy that will be studied in details in order to proceed with a fine analysis.

An additional step in our analysis concerns, the search for the most competent Dynamic Selection techniques related to the chosen pool generation strategy in the first test. Indeed, another Friedman and Bonferoni-Dunn post hoc tests are administered to show the average ranks between the techniques and therefore pick the relevant ones, ideally from both DES and DCS approaches.

The last step of our approach consists in narrowing down the analysis. To do so, a pairwise sign test (Demšar, 2006) is used, based on the number of wins, ties and losses, of the chosen DS techniques for the proposed pool generation method, compared with the performances of the same DS techniques using SGH and Bagging as ensemble generation techniques. The goal of this analysis is to see which DS techniques are more suitable for the proposed generation techniques and on which kinds of classification problems.

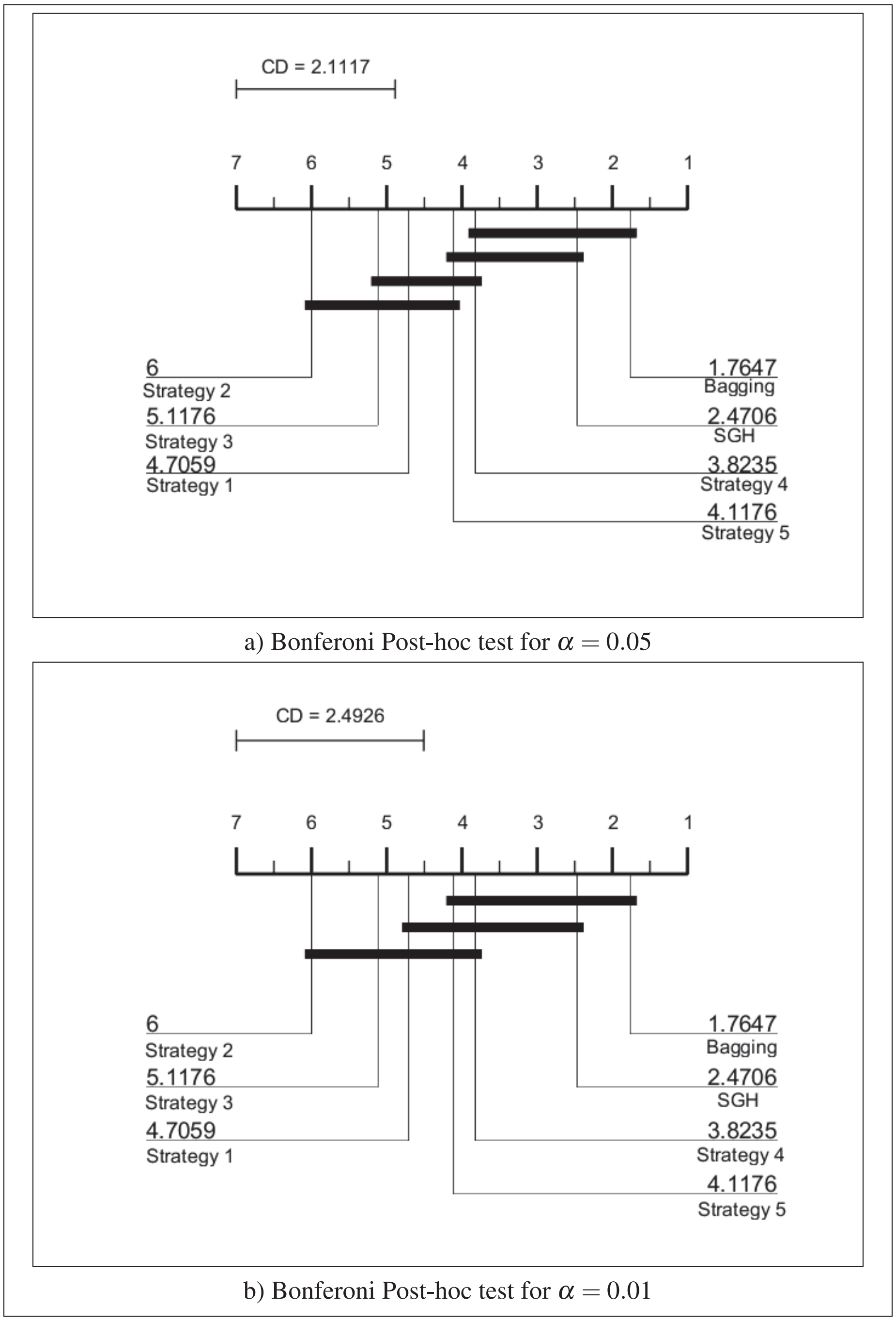
Global comparison: Most suitable strategy

Figure 3.2 shows the critical difference diagram; the techniques in which the difference in average rank is lower than the critical difference are considered as statistically equivalent and hence, they are connected by a bar.

In fact, we can see from Figure 3.2 (a), for a critical value of $\alpha = 0.05$, the proposed strategy 4 and the two baselines are connected with a bar, which means that they are statistically equivalent. On the other hand, Figure 3.2(b) shows the global critical difference for a value of $\alpha = 0.01$ which adds strategy 5 to the most performing methods of this batch of comparison.

From this first round of visual comparison, we observe that strategy 4 is the most promising, as it presents a **global** equivalence in terms of performance for DS techniques compared to the DS techniques results, under the two state of the art ensemble methods.

Now, **when does our method outperform the state of the art?** The next subsections, provide narrow statistical comparisons attempting to answer the question.



a) Bonferoni Post-hoc test for $\alpha = 0.05$

b) Bonferoni Post-hoc test for $\alpha = 0.01$

Figure 3.2 Bonferroni-Dunn post hoc test for a critical value of $\alpha = 0.05$ (on top) and $\alpha = 0.01$ (on the bottom). The strategies in which the difference in average rank is lower than the critical value are connected with a bar.

Strategy 4: The most relevant DS techniques

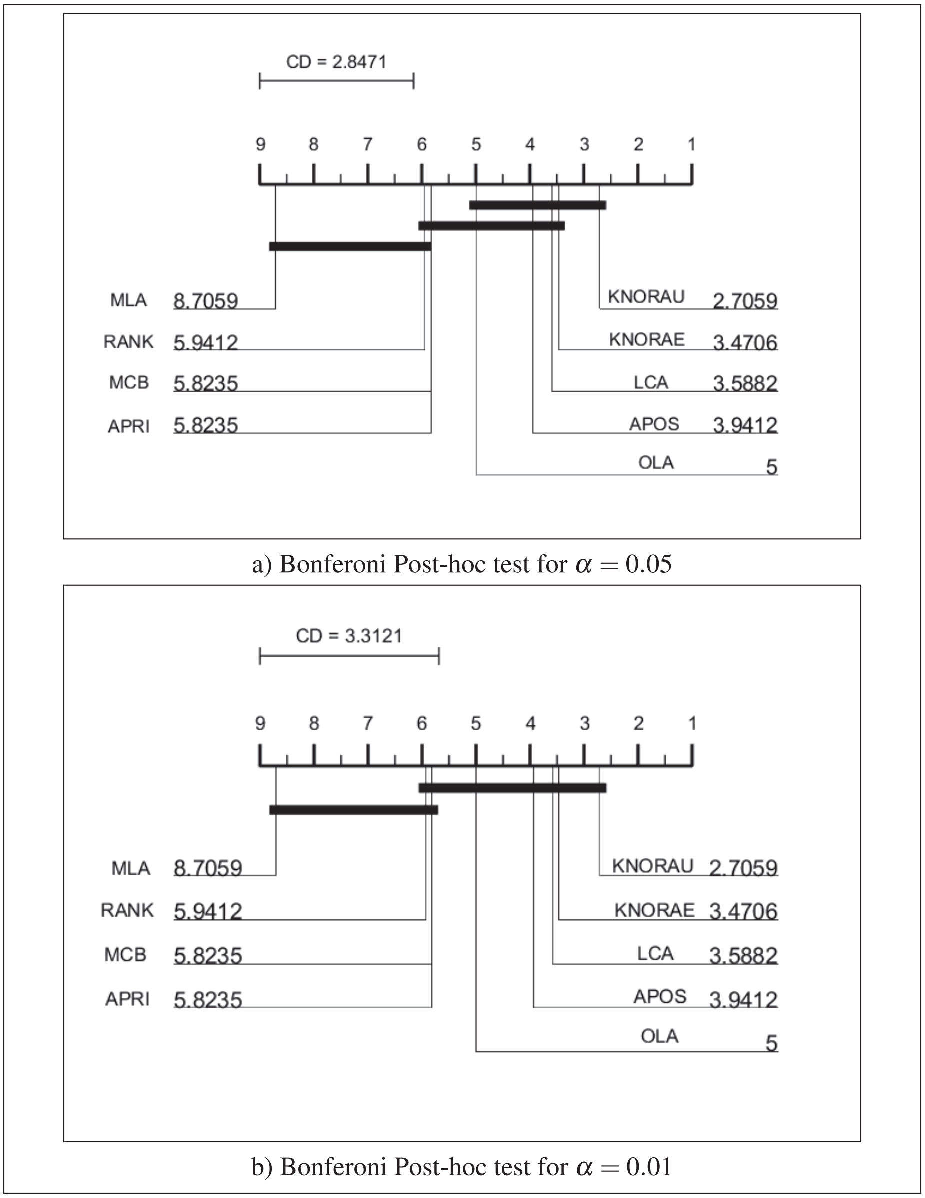
To provide a finer analysis, we conducted again the Friedman average rank test and the Bonferoni-Dunn post hoc test on the pool generation method provided by strategy 4, comparing between the different DS techniques.

The Friedman test provides the average rank between the methods given a critical difference according to the critical values of α so we can proceed with Bonferoni-Dunn post hoc test. The technique providing the lowest average rank is the one presenting the highest results. Besides, the Dynamic Selection techniques in which the difference in average rank is lower than the critical difference are considered as statistically equivalent and hence are illustrated by being connected by a bar.

For $\alpha = 0.05$ represented by Figure 3.3 (a), the critical difference $CD = 2.547$. We see that for strategy 4, the DES method KNORA-U provides the best results for all the datasets. It is statistically equivalent to the DES method KNORA-E and the DCS techniques LCA, Aposteriori (APOS) and OLA. Followed on the other side by the other DS techniques ranking as follows Apriori (APRI), MCB, Rank and Lastly MLA.

For the other critical value of $\alpha = 0.01$, the rank in terms of the best performances remains the same as previously (KNORA-U is ranked first, followed by KNORA-E, LCA, Aposteriori, Apriori, MCB, Rank and MLA). Except that the value of the critical difference is different ($CD = 3.3121$); this includes the DCS techniques Apriori (APRI) and MCB in the category of equally performing DS techniques. We observe in the overall that both the DES techniques perform better than the rest of the DCS techniques for this strategy.

This being exposed, we choose the first ranking four Dynamic Selection techniques to pursue the analyses for our proposed generation method conducted using strategy 4. It means that the DES techniques KNORAU and KNORAE, as well as the DCS techniques LCA and Aposteriori (APOS) will be called for further analyses in comparison with the results of the same DS techniques under the two baseline pool generation techniques (Bagging and SGH). We didn't include OLA to have 2 DS techniques from each category.



a) Bonferoni Post-hoc test for $\alpha = 0.05$

b) Bonferoni Post-hoc test for $\alpha = 0.01$

Figure 3.3 Bonferroni-Dunn post hoc test for a critical value of $\alpha = 0.05$ (a) and $\alpha = 0.01$ (b). The DS techniques in which the the difference in average rank is lower than the critical value are connected with a bar.

When does strategy 4 outperform the baselines in terms of DS results?

From the previous statistical Bonferoni Post-hoc test, we saw that globally, compared to the chosen baselines Bagging and the (SGH), the local selection mechanism that is well suited for our research question (generating classifiers that are locally competent, that pass by the local region and are adapted to DS) is **strategy 4**. This strategy generates at least one classifier per indecision region that can **identify** a **maximum number** of pairs of samples from different classes (**frienemies**).

The reason behind this choice resides in the multiple tests conducted above, while following the recommendation of Cruz *el al.* (Cruz *et al.*, 2019), where they define the ideal classifier, as *"a classifier that could perfectly distinguish between the pairs of frienemies"*. Therefore, our heuristic that presents the local selection strategy tightens its choice to the classifier that maximizes the recognition of the samples in the neighborhood belonging to different classes, is the most suited to the research question.

Given this information, how well did the DS techniques perform with the different pool generation schemes?.

To answer this question, we conducted a pairwise analysis based on the Sign-rank test from (Demšar, 2006) that computes the number of wins, ties and losses obtained by each of the four DS techniques with a pool generated by strategy 4 compared to the same DS methods with a pool generated by Bagging and SGH.

The null hypothesis, H_0, meant that both pool generation approaches obtained (Strategy 4 Vs SGH and Strategy 4 Vs Bagging, respectively) obtained statistically equivalent results. A rejection to H_0 means that the classification performance obtained by the DS technique in the proposed scheme was significantly better at a significance level defined by α.

For this test, the null hypothesis, H_0, is rejected when the number of wins is greater than or equal to a critical value n_c. This value is calculated in equation (3.1), where n_{exp} is the number of experiences conducted (17 in our cases). We considered three levels of significance: $\alpha = \{0.10, 0.05, 0.01\}$.

$$n_c = \frac{n_{exp}}{2} + z_\alpha \frac{\sqrt{n_{exp}}}{2} \tag{3.1}$$

Figure 3.4 shows a pairwise comparison between the performances of DS techniques using Strategy 4 and the for the same methods, achieved by calculating the numbers of wins, ties and losses.

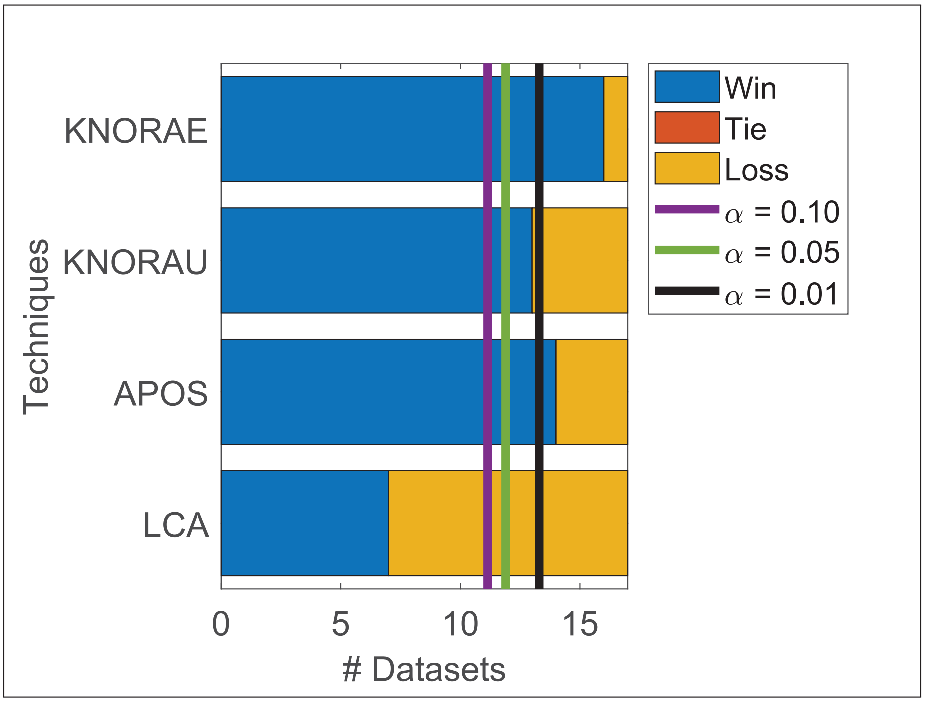


Figure 3.4 Strategy 4 compared with SGH using the different DS techniques. The colored lines (left to right) illustrate the critical values n_c considering significance levels of $\alpha = \{0.10, 0.05, 0.01\}$, respectively

Compared to SGH for the different DS techniques, we can see that Strategy 4 outperforms the for KNORA-E at all the significance levels (ie, for $\alpha = \{0.10, 0.05, 0.01\}$. It presents as well, a significant number of wins at all the values of α. For Aposteriori (APOS), the same conclusion is being drawn in terms of the number of wins of our generation strategy compared to the baseline. Only LCA did not present a significant number of wins, nevertheless, it remains statistically equivalent to the baseline for this case.

Figure 3.4 shows that overall, our proposed strategy 4 statistically outperforms the baseline for KNORA-E, KNORA-U and Aposteriori(APOS) and this for all the levels of significance.

On the other hand, Figure 3.5 shows a pairwise comparison between the performances of DS techniques using Strategy 4 and Bagging for the same methods, achieved by calculating the numbers of wins, ties and losses.

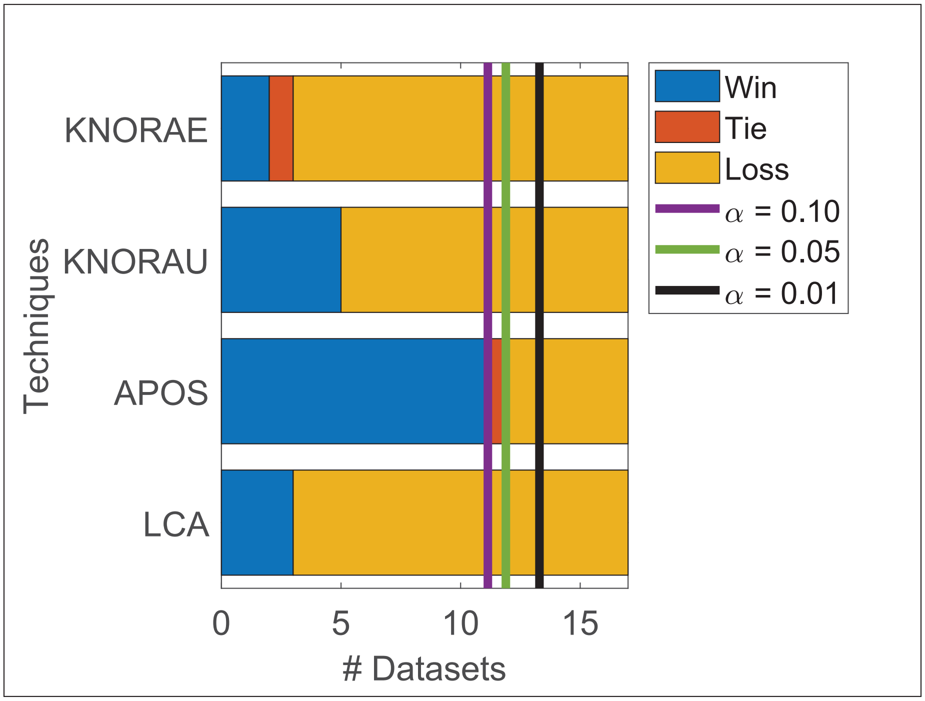


Figure 3.5 Strategy 4 compared with Bagging using the different DS techniques. The colored lines (left to right) illustrate the critical values n_c considering significance levels of $\alpha = \{0.10, 0.05, 0.01\}$, respectively

We see that for our proposed strategy 4, according to Figure 3.5, Aposteriori (APOS) outperformed the Aposteriori results given by a pool generated by the baseline Bagging at a significance level of $\alpha = 0.1$. It also presented a tie with the baseline for $\alpha = 0.05$.

In the overall, our proposed strategy 4 statistically outperforms significantly the baseline with Bagging for Aposteriori (APOS) for a level of significance $\alpha = 0.1$ and a tie with Bagging for $\alpha = 0.05$. It did not statistically present a significant number of wins for the rest of the methods. However, they remain statistically equivalent for a certain significant level.

The statistical analysis showed us somehow, the behavior of our most performing Pool generation strategy in terms of DS results, compared to the DS results given by the two different pool generation techniques that constitute our baseline methods.

The baseline ensemble generation methods are characterized as follows: (1) Bagging is the usual Ensemble generation approach for the DS technique (Cruz *et al.*, 2018) that covers the features space. However, It remains a non-informed method based on resampling with replacement (Breiman, 1996) that guarantees a certain diversity. On the other hand, SGH (2) is a method that guarantees a value of the oracle of 100%, it is intuitive and provides a small size pool of classifiers. However, none of these methods take into account the aspect of locality when creating the predictors and our proposed methods focuses on this, to create the pool of classifiers.

Compared to SGH, our method performed very well by presenting significantly better results on almost all the DS techniques studied and when it did not, it showed equivalence. Whereas in comparison with Bagging, the statistical outperformance of our approach was only significant for Aposteriori. This, dragged our attention on what could be the possible explanations for such results.

Given that our system fully depends on the local information in the Region of Competence and this local information relies on the number of examples in *DSEL* for each sample that we treat; an interesting path was to compute the Ratio of the number of Samples on the number of Features as $SFR = \frac{number of samples}{number of features}$.

Table 3.2 represents this ratio for each dataset ranked from the lowest to the highest. On the columns is represented the average result of APOS for Bagging, SGH and DSPG for Strategy 4. We can see directly from the table that for the Dataset "**Sonar**" Bagging and SGH win against DSPG for value of $SFR = 3.47$.

On the contrary, for a value of $SFR \geq 3.47$, the proposed method either dominates or is statistically equivalent to the state of the art's method. It is expected, as our technique focus solely on local information gathered in $DSEL$, where for these kinds of datasets, we don't necessarily dispose of enough information to create locally competent classifiers visible by the DS techniques.

Table 3.2 Mean of the accuracy of APOS for Bagging, SGH and DSPG (Strategy 4) given the number of example on the number of features ratio (SFR)

Dataset	SFR	Bagging	SGH	DSPG
Sonar	**3.47**	**70.87**	69.23	66.92
Ionosphere	9.26	80	77.27	**82.39**
Laryngeal1	13.31	**82.26**	80.38	**82.26**
Weaning	17.76	73.95	72.37	**74.47**
Breast	18.93	95.35	95.14	**95.70**
Heart	20.77	85.88	84.85	**86.62**
German	50	69.08	69.28	**71.00**
Vertebral	51.67	**82.95**	80.77	81.92
Liver	57.5	**64.07**	62.09	63.49
ILPD	58.3	67.19	66.85	**68.29**
Pima	96	72.92	72.03	**75.1**
Haberman	102	70.13	69.41	**72.63**
Blood	187	70.61	**71.78**	71.25
Mammographic	192.2	79.11	**79.45**	78.8
Lithuanian	500	96.03	96.13	**96.87**
Monk2	720	**83.89**	81.67	83.43
Adult	3488.71	85.09	85.55	**87.17**

3.3 General discussion

In this chapter, we presented an overview of the DS results of the proposed strategies, as well as the results of some baselines from the literature (Bagging, and).

Compared to SGH, our strategy 4 presents higher performances in most of the DS techniques chosen to address the analysis. Whereas for Bagging, our method presented a significant number of wins for one DS technique and slightly inferior number of loss for the other ones. This lead us to, further investigate the reasons behind this difference in the performances by calculating the ratio of the number of samples on the number of features.

The trend shows promise in terms of pursuing this research direction of creating locally competent classifiers for the context of Dynamic Selection. Below, are some relevant points we could observe:

The most suitable guide for local selection of classifiers

As one of our objectives was to create a pool of locally competent classifiers adapted to the context of DS, we generate several strategies to guide the creation of the pool. The most suitable strategy appeared to be the one exploiting the **maximization** of the well classification of the frienemies within RoC, when compared to the other strategies and the ones in the literature. Indeed, each strategy had its motivations, but the chosen one had not only covered the indecision region, but also maximized the local performance by maximizing the number of well classified frienemies.

The impact of the locality on the number of classifiers

In this work, we assumed that a local classifier is a hyperplane that crosses the region of competence.

As one of our research questions concerns the creation of a pool of locally competent classifiers, a classifier has been created for every hard sample of the dataset to ensure the local competency and the locality in the indecision region. Therefore, the number of classifiers obtained is proportional to the number of hard samples within the datasets. Appendix II shows the average number of classifiers generated for Aposteriori for the different strategies suggested in this research study (169 classifiers in average for strategy 4), compared to the number of classifiers generated in SGH (usually less than 5 perceptrons)

How does the proposed methodology stand out from the literature?

Our proposed method stands out from the literature by exploiting the concept of locality to generate a pool of classifiers in the context of DS, as opposed to the baseline methods, which use global information to cover the features space.

In terms of results, Maximizing the frienemies recognition while generating a pool of classifiers had a positive impact on the results of the DS techniques in comparison with the baselines. It presented a significant number of wins against the baseline that uses SGH for almost all the DS techniques. The reason behind this result is that the pool generated by SGH covers the features space globally as opposed to the proposed method. Moreover, it generates a few number of classifiers (less than 5 in average, Appendix II) which makes it challenging to achieve excellent performances when dealing with hard samples found in indecision regions, compared to DSPG.

As for the DS results with Bagging, our methods outperformed on the Aposteriori DS techniques and had equivalent and slightly inferior results on the rest of the techniques given the 17 datasets. In terms of features space coverage, Bagging meets the requirements because it generates classifiers for different areas of the features space. However, it remains a non-informed global generation technique that does not take into account local information as opposed to DSPG.

On the other hand, given the poor information brought by *DSEL* in the cases where there is a low ratio of $SFR = \frac{number\,of\,samples}{number\,of\,features}$, added to the full dependency of the proposed system on the local information in the Region of Competence; the DS techniques struggle to localize the most competent classifiers in generalization, even if we believe the truest ones were created.

The reason behind this struggle resides in the fact that, there is no link between the classifiers generated locally during training with the samples located in similar areas in generalization. This phenomenon lead again to a global use of the classifiers generated locally by our method.

This chapter was very insightful in terms of the efficiency of the proposed system when compared to the baselines. It highlighted and confirmed its pros and brought up some issues that need to be taken into consideration for a further pursue of the research question on the locality for Dynamic Selection.

Further investigations need to be lead. The recommendations given the announced drawbacks of the system, will be discussed in the next chapter.

CONCLUSION AND RECOMMENDATIONS

In this work, the problem statement expressed the need of creating an ensemble generation method that is adapted to the Dynamic Selection context. Therefore, we had to study many aspects of the dynamic selection scheme through the related work (chapter 1), and explore many ensemble generation methods. Throughout the process, we conducted an analysis (conference paper (Cruz *et al.*, 2017)) (Appendix I) that showed us the areas of the efficiency of Dynamic Selection technique regarding the complexity of the data.

Throughout this thesis, we attempted to answer the following research questions:

- **Can we create locally competent classifiers that are adapted to the DS requirements in terms of local information?**

Over the proposed methodologies presented in chapter 2, we created classifiers that separate the frienemies (*samples located in the same region of competence, from different classes*) within the regions of competence; taking into consideration the proportion of the samples in the indecision regions. We remind that a locally competent classifier is assumed to be a hyperplane that crosses the region of competence for a specific query, and is competent in the classification of the samples within the neighborhood.

- **What are the possible guides to lead to the creation of a pool of locally competent classifiers?**

For this case, we presented a novel classifier generation method based on several strategies that generates local classifiers and takes into consideration local criteria in chapter 2. The main characteristics of these strategies are the creation of several hyperplanes that separate the frienemies (TP), within the region of competence, and use (1) DCS techniques to choose the most suitable classifier within that RoC, (2) KNORA-E as a local oracle to find the set of the most competent classifiers or (3) maximizing the distinction between the frienemies in RoC. The **DSPG** strategy focuses on the local coverage of the indecision regions of competence. This means that, for every sample belonging to a certain neighborhood, there is at least one classifier that is competent enough to join the **DSLP**. By the end of the pool generation, the features space is covered in indecision regions. A detailed case study on a synthetic dataset was provided to understand the method. The results showed that the most suitable guide to lead the creation of a pool of locally competent classifier is maximizing the frienemies distinction.

- **How can we jointly use $K-NN$ and DS in generalization based on the instance hardness measure?**

Following the recommendations in (Cruz *et al.*, 2017) (Appendix I), the joint use of the $K-NN$ and DS depending on the level of hardness of the instance is promising as a classification system, exploits the advantages of the two strategies. As it allows the $K-NN$ to operate fast on the samples judged to be easy to classify and leaves the DS techniques to focus on the rest of the samples. For our case, we considered an instance to be easy if its region of competence is homogeneous.

The results of this study in chapter 3, were comparable to the results of the baseline methods. Compared to SGH in DS results, our methods presented a significant boost in terms of performances in generalization for most of the DS techniques. As for Bagging, our methods provided a statistical improvement on one of the DS techniques and presented situations of equivalence and slight inferiority. This lead us to observe that for some high dimension and small size datasets the results were the poorest compared to ones in the literature, given the poor information brought by *DSEL*. This situation lead to a clear struggle of some DS techniques to find the most competent classifiers due to the insufficient representation of the data; added to the fact that the use of these locally generated competent classifiers is used globally by the DS techniques; as there is no link between the classifiers generated and their use in generalization.

This work provided us with several interesting insights concerning the integration of local information in constructing an ensemble of classifiers. The trend shows promise in terms of pursuing this research direction. Naturally, there is still room for improvement for such approaches.

Future work

In this part, we present the recommendations that we suggest for this research problem, towards local classifier generation for dynamic selection.

- Conduct a deep investigation of the correlation between the ratio of the number of samples and the number of features (SFR) to improve the local pool generation.
- Use prototype generation techniques in *DSEL* for datasets with high dimensionality and small amount of data, given that that the generation method is fully dependent on the information provided in *DSEL*.
- Create a heuristic that forces the DS technique to choose the classifier(s) that has (have) been generated for that specific Region of Competence .
- Enlarge the definition of easy samples by including the ones having lower instance hardness rate $IH < 0.42$ as suggested in (Cruz *et al.*, 2017) to be treated by the K-NN classifier.
- Construct a new online classifier generation/selection system that operates in the indecision regions as follows: for each hard test sample, create a Temporary Pool (TP) within the RoC. Then rely on the decision of the classifier that separates between the maximum of frienemies (as in strategy 4) or more than 1 classifiers (as in strategy 5) and conduct a majority voting amongst them for predicting the label of the query.
- Extend and enlarge the work to multi-class datasets.

APPENDIX I

COMMUNICATION PRESENTED IN IPTA 2017

This appendix contains complementary information to this research. The communication was presented in International Conference on Image Processing Theory, Tools and Applications (IPTA 2017). In this paper, we state why and when Dynamic Selection obtains higher classification performance than the K-NN classifier.

Dynamic Ensemble Selection VS K-NN: why and when Dynamic Selection obtains higher classification performance?

Rafael M. O. Cruz[1], Hiba H. Zakane[1], Robert Sabourin[1] and George D. C. Cavalcanti[2]
[1]École de Technologie Supérieure - Université du Québec, Canada
e-mail: rafaelmenelau@gmail.com, zakanehiba@gmail.com, Robert.Sabourin@etsmtl.ca
[2]Centro de Informática - Universidade Federal de Pernambuco, Brazil
e-mail: gdcc@cin.ufpe.br

***Abstract*— Multiple classifier systems focus on the combination of classifiers to obtain better performance than a single robust one. These systems unfold three major phases: pool generation, selection and integration. One of the most promising MCS approaches is Dynamic Selection (DS), which relies on finding the most competent classifier or ensemble of classifiers to predict each test sample. The majority of the DS techniques are based on the K-Nearest Neighbors (K-NN) definition, and the quality of the neighborhood has a huge impact on the performance of DS methods. In this paper, we perform an analysis comparing the classification results of DS techniques and the K-NN classifier under different conditions. Experiments are performed on 18 state-of-the-art DS techniques over 30 classification datasets and results show that DS methods present a significant boost in classification accuracy even though they use the same neighborhood as the K-NN. The reasons behind the outperformance of DS techniques over the K-NN classifier reside in the fact that DS techniques can deal with samples with a high degree of instance hardness (samples that are located close to the decision border) as opposed to the K-NN. In this paper, not only we explain why DS techniques achieve higher classification performance than the K-NN but also when DS should be used.**

***Keywords*— Ensemble of classifiers, Dynamic ensemble selection, K-nearest neighbors, Instance hardness.**

I. Introduction

One of the most promising MCS approaches is Dynamic Selection (DS), in which the base classifiers[1] are selected on the fly, according to each new sample to be classified. DS has become an active research topic in the multiple classifier systems literature in the past years. This is due to the fact that more and more works are reporting the superior performance of such techniques over traditional combination methods, such as Majority Voting and Boosting [1], [2], [3], [4], [5]. DS techniques work by estimating the competence level of each classifier from a pool of classifiers. Only the most competent, or an ensemble containing the most competent classifiers, is selected to predict the label of a specific test sample. The rationale for such techniques is that not every classifier in the pool is an expert in classifying all unknown samples; rather, each base classifier is an expert in a different local region of the feature space [6].

In dynamic selection, the key is how to select the most competent classifiers for any given query sample. The competence of the classifiers is estimated based on a local region of the feature space where the query sample is located, called region of competence. This region is usually defined by applying the K-Nearest Neighbors technique to find the neighborhood of this query sample. Then, the competence level of the base classifiers is estimated, considering only the samples belonging to the region of competence according to any selection criteria; these include the accuracy of the base classifiers in this local region [7], [8], [9] or ranking [10] and probabilistic models [3], [11]. The classifier(s) that attained a certain competence level is(are) selected.

Several works pointed out that the performance of DS techniques is very sensitive to the definition of the region of competence [12], [13], [14]. If there is a noise in the defined neighborhood of the query sample, the DS systems are more likely to fail. Moreover, the use of different K-NN approaches for the definition of the regions of competences can significantly change the performance of DS methods [15].

As the competence of the base classifiers are heavily dependent on the K-Nearest Neighbors for the definition of the local regions, one question arises: Why do we use dynamic selection instead of simply applying the K-NN classifier? Moreover, in which scenario the use of DS brings benefits over the K-NN? To the best of our knowledge, there is no comparison between both classification approaches in the DS literature. Hence, the objective of this paper is to perform an analysis comparing the classification results of DS techniques and the K-NN classifier. In particular, the following points are investigated:

1) Do DS techniques achieve higher classification performance than the K-NN?
2) Why does DS present better classification accuracy than K-NN even though the same neighborhood is considered for both techniques?
3) When should DS be used for classification instead of K-NN?

Experiments are carried out using 18 state-of-the-art DS technique over 30 classification datasets. We demonstrate that

[1] The term base classifier refers to a single classifier belonging to an ensemble or a pool of classifiers.

978-1-5386-1842-4/17/$31.00 © 2017 IEEE

not only DS techniques achieves significantly better results, but we also demonstrate in which scenarios DS techniques can improve the generalization performance over the K-NN classifier.

This paper is organized as follows: Section II presents the related works on dynamic selection. Section III addresses the experiments conducted on state-of-the-art DS techniques. Conclusion and future works are presented in the last section.

II. Dynamic selection

Dynamic selection techniques consist, based on a pool of classifiers C, in finding a single classifier c_i, or an ensemble of classifiers $C' \subset C$, that has (or have) the most competent classifiers to predict the label for a specific test sample, $\mathbf{x}_q$. The most important component of DES techniques is how the competence level of the base classifier is measured, given a specific test sample $\mathbf{x}_q$. This is a different concept from static selection methods [16], [17], in which the Ensemble of Classifiers (EoC), $C^{'}$, is selected during the training phase, according to a selection criterion estimated in the validation dataset, and is used to predict the label of all test samples in the generalization phase.

In dynamic selection, the classification of a new query sample normally involves three phases:

1) The definition of the region of competence; that is, how to define the local region surrounding the query, $\mathbf{x}_q$, in which the competence level of the base classifiers is estimated
2) The selection criteria used to estimate the competence level of the base classifiers, e.g., Accuracy, Probabilistic, and Ranking
3) The selection mechanism that chooses a single classifier (DCS) or an ensemble of classifiers (DES) based on their estimated competence level

The most common method to define the regions of competence is by using the K-NN technique, to get the neighborhood of the test sample [7], [4], [8], [18], [19], [20], [21], [5], [11], [9], [10], [22]. The set with the K-Nearest Neighbors of a given test sample $\mathbf{x}_q$ is called region of competence, and is denoted by $\theta_q = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$. Many works pointed out that the definition of this region of competence is of fundamental importance to DS methods, as the performance of all DS techniques is very sensitive to the distribution of this region [15], [23]. The samples belonging to θ_q are used to estimate the competence of the base classifiers, for the classification of $\mathbf{x}_q$, based on various criteria, such as the overall accuracy of the base classifier in this region [7], ranking [10], ambiguity [24], oracle [4] and probabilistic models [11]. In any case, a set of labeled samples, which can be either the training or validation set, is required for the definition of the local regions. This set is called the dynamic selection dataset (DSEL) [25].

After the competence level of the base classifiers are estimated, the most competent one or an ensemble containing the most competent classifiers, to predict the label of $\mathbf{x}_q$ is(are) selected. For instance the Overall-Local-Accuracy (OLA) [7] and Multiple Classifier Behavior (MCB) [20] techniques select only the classifier that achieved the highest competence level in the neighborhood, while the K-Nearest Oracles techniques (KNORA) [4] and the Dynamic Ensemble Selection-Performance (DES-P) [11], and META-DES [5] select an EoC containing the most competent classifiers.

As the neighborhood of the query sample is not used directly to predict its label, but rather to estimate the competence level of the base classifiers. This brings benefits when dealing with samples located in an indecision region, i.e., which are located in areas surrounding classes boundaries [26]. When the query is located in such a region, the majority of its K-Nearest Neighbors may belong to a different class, which can lead to bad predictions. Moreover, samples located in indecision regions are often misclassified by other pattern recognition techniques since they are usually associated with a high degree of instance hardness [27].

However, DS techniques can still predict the correct label for such samples as long as there exists at least one base classifier that is competent locally. In other words, a classifier that can correctly classify samples belonging to different classes in the indecision regions. For example, Figure 1 shows an example of an indecision region. The query sample $\mathbf{x}_{query}$, belongs to the class 1 (red square). Since the majority of its neighbors comes from the class 2 (blue circle), a K-NN classifier, considering this whole neighborhood, would misclassify the query sample.

Using dynamic selection it is possible to predict the correct label of such sample as long as there are base classifiers that cross this indecision region. For instance, consider the system consisting of four base classifiers as shown in Figure 1 (b). If we apply the Overall-Local-Accuracy (OLA) technique [7], the classifier c_3 would be selected, since it obtained a 100% accuracy for the local region. The other base classifier that predicted the correct label is c_1 yet, it does not cross the region of competence, knowing that it achieves a level of competence of 0.33 which is lower than classifiers c_2 and c_4 with 0.85 and 0.57 respectively. Consequently, using dynamic selection it is possible to give the correct prediction for this sample as long as there is at least one base classifier that obtains a high competence level in the local region.

Thus, our hypothesis is that DS techniques outperform the K-NN classifier since it can better deal with samples that are located in indecision regions. This hypothesis is evaluated in the next section.

III. Experiments

The comparative study was performed using a test bed composed of 30 classification problems proposed in [5]. The key features of the datasets are presented in Table 1. For each dataset, the experiments were carried out using 20 replications. For each replication, the datasets were randomly divided on the basis of 25% for training, $\mathcal{T}$, 50% for the dynamic selection dataset, $DSEL$, and 25% for the generalization set, $\mathcal{G}$. The divisions were performed while maintaining the prior probabilities of each class. For the K-NN classifier, $DSEL$

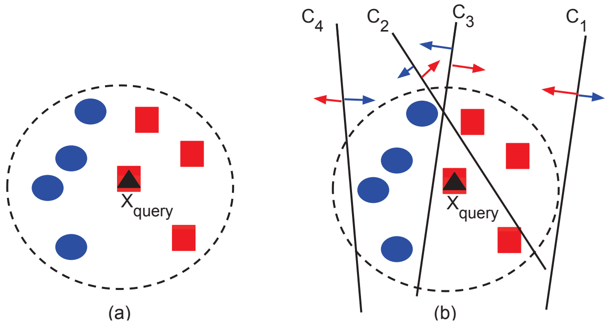


Fig. 1. Example of a query sample located in an indecision region. (a) The estimated region of competence with K = 7. (b) Decision border of different base classifiers with the arrows pointing to the regions of both classes. As the majority of its neighbors belong to a different class, the K-NN classifier would make the wrong prediction. However, DS techniques can still make the right decision if the DS method selects the base classifier that is competent locally (c_3).

Table 1. Summary of the 30 datasets used in the experiments [Adapted from [5]].

Database	No. of Instances	Dimensionality	No. of Classes	Source
Adult	48842	14	2	UCI
Banana	1000	2	2	PRTOOLS
Blood transfusion	748	4	2	UCI
Breast (WDBC)	568	30	2	UCI
Cardiotocography (CTG)	2126	21	3	UCI
Ecoli	336	7	8	UCI
Steel Plate Faults	1941	27	7	UCI
Glass	214	9	6	UCI
German credit	1000	20	2	STATLOG
Haberman's Survival	306	3	2	UCI
Heart	270	13	2	STATLOG
ILPD	583	10	2	UCI
Ionosphere	315	34	2	UCI
Laryngeal1	213	16	2	LKC
Laryngeal3	353	16	3	LKC
Lithuanian	1000	2	2	PRTOOLS
Liver Disorders	345	6	2	UCI
MAGIC Gamma Telescope	19020	10	2	KEEL
Mammographic	961	5	2	KEEL
Monk2	4322	6	2	KEEL
Phoneme	5404	6	2	ELENA
Pima	768	8	2	UCI
Satimage	6435	19	7	STATLOG
Sonar	208	60	2	UCI
Thyroid	215	5	3	LKC
Vehicle	846	18	4	STATLOG
Vertebral Column	310	6	2	UCI
WDG V1	5000	21	3	UCI
Weaning	302	17	2	LKC
Wine	178	13	3	UCI

was merged with the training data. As a result, all methods were trained using the same amount of data available, while the distribution of the test set remained the same. The pool of classifiers C was composed of 100 Perceptrons generated using the Bagging technique. The same pool of classifiers was used for all DS techniques. Moreover, the size of the region of competence (neighborhood size) K was equally set at 7 for all techniques since it presented the best classification performance according to [4], [5].

The analysis is conducted using 18 state-of-the-art DS techniques, eight DCS and ten DES techniques. For DCS, the following techniques were evaluated: Local Class Accuracy (LCA) [7], Overall Local Accuracy (OLA) [7], Modified Local Accuracy (MLA) [8], Modified Classifier Ranking (RANK) [10], [7], Multiple Classifier Behavior (MCB) [20], A Priori [18], [19], A Posteriori [18], [19] and the Dynamic Selection on Complexity (DSOC). For dynamic ensemble selection, the following techniques were considered: K-Nearest Oracles Eliminate (KNORA-E) [4], K-Nearest Oracles Union (KNORA-U) [25], Randomized Reference Classifier (DES-RRC) [28], K-Nearest Output Profiles (KNOP) [25], [29], Dynamic Ensemble Selection Performance (DES-P) [11], Dynamic Ensemble Selection Kullback-Leibler (DES-KL) [11], DES Clustering [9], DES-KNN [9], Meta Learning for Dynamic Selection (META-DES) [5] and META-DES.Oracle [30].

Pseudo-code for the implementation of each method is given in [2], [1]. It is important to point out that 15 out of the 18 DS techniques use the K-NN to define the region of competence, the only exceptions being the DES-RRC, DES-KL and the DES-KMEANS. However, they still use local information in order to estimate the competence level of the base classifiers.

A. Comparison DS vs K-NN

The first analysis conducted in this paper is a comparison between the accuracy obtained by DS techniques and the K-NN classifier. The objective of this analysis is to know whether the use of DS leads to a significant improvement in classification accuracy. For the K-NN classifier, we consider a $K = 7$ (i.e., the same neighborhood size used by the DS techniques) as well as the $K = 1$ which is used as a baseline comparison.

Table 2 shows the average ranking and mean accuracy of each technique considering the 30 classification problems studied. The average ranks were obtained using the Friedman test [31] as follows: For each dataset, the method that achieved the best performance received rank 1, the second best rank 2, and so forth. In case of a tie, i.e., two methods presented the same classification accuracy for the dataset, their average ranks were summed and divided by two. The average rank was then obtained, considering all datasets. The best performing algorithm, considering the 30 classification datasets, was the one presenting the lowest average rank.

All DS techniques presented a better ranking and average accuracy when compared to the 1-NN, and only the MLA technique presented a lower classification accuracy and lower rank than the K-NN using the same neighborhood size (K=7). This is an interesting finding, since the majority of the DS techniques in this study (14 methods) use the K-NN method in the process of estimating the local competence of the base classifiers.

Furthermore, a pairwise analysis was conducted based on the Sign test [32], computed on the number of wins, ties and losses obtained by each DS, compared to the 7-NN (i.e., same neighborhood size). The null hypothesis, H_0, meant that both techniques obtained statistically equivalent results. A rejection in H_0 meant that the classification performance obtained by a corresponding DS technique was significantly better at a predefined significance level α. In this case, the null hypothesis, H_0, is rejected when the number of wins is

Table 2. Overall results

Algorithm	Avg. Rank	Algorithm	Avg. Accuracy
META-DES.O	4.07(3.67)	META-DES.O	83.92(9.13)
META-DES	4.40(3.23)	META-DES	83.24(8.94)
DES-RRC	6.40(5.30)	DES-P	82.26(9.26)
KNORA-U	7.33(4.65)	DES-RRC	82.11(8.76)
DES-P	7.57(4.06)	KNORA-U	81.69(9.82)
DES-KL	8.20(5.43)	DES-KL	81.52(8.77)
KNOP	10.27(4.19)	KNOP	80.81(8.92)
KNORA-E	10.40(4.21)	KNORA-E	80.36(10.75)
LCA	10.80(4.91)	OLA	79.87(10.67)
OLA	11.07(5.23)	DCS Rank	79.69(10.38)
DSOC	11.63(6.17)	DSOC	79.68(9.44)
MCB	11.93(5.39)	LCA	79.57(9.84)
DES-KNN	12.00(4.72)	MCB	79.56(9.70)
A Posteriori	12.17(5.68)	DES-KNN	79.29(10.23)
DCS Rank	12.53(4.53)	A Priori	78.57(11.18)
7NN	12.97(6.32)	DES-KMEANS	78.49(10.40)
DES-KMEANS	13.57(4.26)	A Posteriori	78.14(11.53)
MLA	13.63(5.12)	7NN	77.42(13.06)
A Priori	13.77(4.67)	MLA	77.34(9.78)
1NN	15.30(5.95)	1NN	76.64(11.98)

greater than or equal to a critical value, denoted by n_c. The critical value is computed using Equation 1

$$n_c = \frac{n_{exp}}{2} + z_\alpha \frac{\sqrt{n_{exp}}}{2} \tag{1}$$

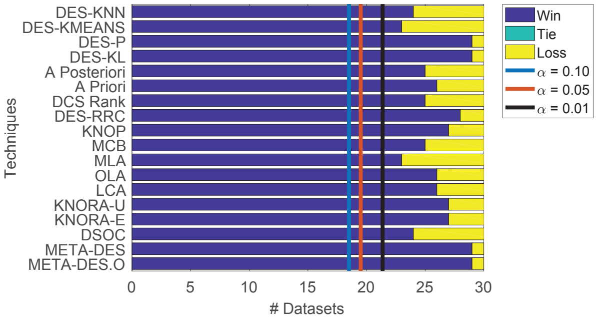


Fig. 2. Pairwise comparison between the results achieved using the different DS techniques and the 1-NN. The analysis is based on wins, ties and losses. The vertical lines illustrate the critical values considering a confidence level $\alpha = \{0.10, 0.05, 0.01\}$.

where n_{exp} is the total number of experiments. We ran the test considering three levels of significance: $\alpha = \{0.10, 0.05, 0.01\}$. Figures 2 and 3 show the results of the Sign test comparing the performance of DS techniques and the 1-NN and 7-NN respectively. The different bars represent the critical values for each significance level.

Compared to the 1-NN, we can see that all DS methods presented a significant number of wins even when the level of significance is reduced to $\alpha = 0.01$. Compared to the 7-NN (i.e., the same neighborhood size as the DS techniques) we can see that at a 0.1 significance level, all DS techniques obtained a significant number of wins. Using an $\alpha = 0.05$, only two DS methods (DS-KMEANS and MLA) did not present a significant number of wins. Moreover, even restricting the test to a significance level of 0.01, we could see that the majority of the DS techniques obtained a significant number of wins. Therefore DS methods present a significant boost in classification accuracy even though they use the same neighborhood as the K-NN.

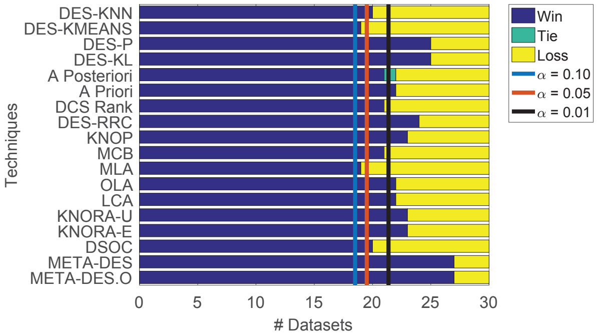


Fig. 3. Pairwise comparison between the results achieved using the different DS techniques and the 7-NN. The analysis is based on wins, ties and losses. The vertical lines illustrate the critical values considering a confidence level $\alpha = \{0.10, 0.05, 0.01\}$.

B. Instance hardness analysis

Instance hardness (IH) measure provides a framework for identifying which instances are hard to classify and also, understand why they are hard to classify [27]. The objective of this experiment is to analyze the performance of DS techniques and the K-NN classifier for dealing with samples with different degrees of instance hardness. Thus, we want to test our hypothesis that DS techniques can better handle samples that are located in indecision regions, and are associated with a higher degree of instance hardness.

The kDisagreeing Neighbors (kDN) is considered, since it presented the highest correlation with the probability that a given instance is misclassified by different classification methods according to [27]. The kDN measure is the percentage of instances in an instance's neighborhood that do not share the same label as itself. Equation 2 shows the kDN measure.

$$kDN(\mathbf{x}_q) = \frac{\mid \mathbf{x}_k : \mathbf{x}_k \in KNN(\mathbf{x}_q) \wedge t(\mathbf{x}_k) \neq t(\mathbf{x}_q) \mid}{K} \tag{2}$$

where $KNN(\mathbf{x}_q)$ is the set of K nearest neighbors of $\mathbf{x}_q$, and $\mathbf{x}_k$ represents an instance in this neighborhood. $t(\mathbf{x}_q)$ and $t(\mathbf{x}_k)$ represents the target class of the instances $\mathbf{x}_q$ and $\mathbf{x}_k$ respectively.

In this work, we considered a neighborhood size $K = 7$ for the estimation of the kDN, which is the same neighborhood sized used for the DS techniques as well as the K-NN classifier.

We rank the testing instances of all datasets according to their level of IH. Then, the samples were divided into 8 groups (given that $K = 7$) with the possible configurations of IH (starting from IH = 0, when the whole neighborhood agrees with the class of the test sample, up to IH = 1, when the whole neighborhood disagrees with the label of the test sample).

Then, the classification accuracy of each DS technique and the K-NN are evaluated for each specific group of instances. The results of the DS techniques and K-NN according to the hardness level of the instance are presented in Figure 4. For the sake of simplicity, we considered only the top six DS algorithms. Moreover, only the 7-NN was considered since it outperformed the 1-NN. Based on this analysis, we can see that DS methods achieve higher performance for samples with a high degree of instance hardness. When the IH level is low (IH < 0.4), the K-NN method presents the best result. However, we can see a huge drop in classification accuracy when the IH level increases.

The accuracy of the K-NN for the samples with IH = 0.7 is around 5%, while the best DS techniques obtain an accuracy higher than 50% for such instances (META-DES, META-DES.O and KNORA-U). Moreover, for an IH higher than 0.71 the classification accuracy of the K-NN is equal to zero, while the best DS technique obtained a much higher classification accuracy for such samples. Hence, the reasons behind the outperformance of DS techniques over the K-NN method can be explained by the fact that DS techniques can better deal with samples that are associated with a high degree of instance hardness.

We can clearly see that DS methods outperform the K-NN for the classification of samples associated with a high degree of instance hardness. This is due to the fact that a high IH value means that the majority of the samples in the neighborhood of the query instance come from a different class. Therefore, the K-NN classifier cannot predict the correct label. However, when using DS techniques, it is possible to achieve the correct prediction for such instances as long as there is at least one base classifier or a few that crosses the neighborhood of the query sample (as shown in Figure 1). This result explains why DS techniques often outperform the K-NN classifier, even though the same neighborhood size is considered by both techniques.

Hence, we are able to answer two questions posed in the paper: The reasons why DS techniques present a better performance than the K-NN is due to the fact that DS techniques can deal with samples with a high degree of instance hardness. Moreover, DS techniques should be used for the classification of instances that are associated with a high degree of instance hardness (samples that are located close to the decision border), while the K-NN should be used for the classification instances associated with a low degree of instance hardness (e.g., IH < 0.4). Moreover, for all sample associated with a high degree of IH (IH > 0.4), that were correctly classified by a DS algorithm, there was at least one base classifier in the pool crossing the region of competence (i.e., which could predict the correct label for samples of different classes).

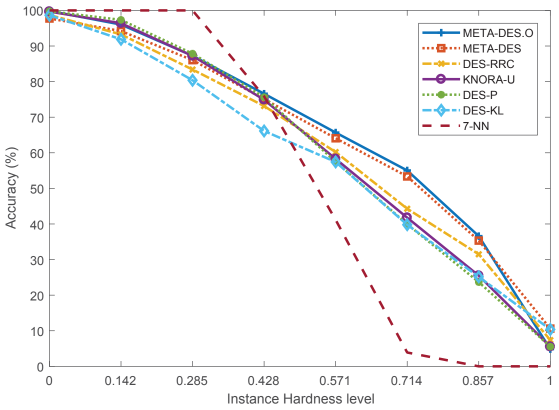


Fig. 4. Performance of DS techniques and K-NN according to the hardness level of an instance considering all 30 classification datasets.

Thus, DS techniques are able to correctly classify instances that are associated with a high degree of instance hardness as long as they can select the base classifiers that are competent locally. For such a condition to be satisfied, it is required that there is at least one base classifier crossing the decision border in the local region of the query sample (as shown in Figure 1). The classifier should also obtain a high local performance in order to be selected by the corresponding dynamic selection technique. Moreover, it would be preferable to guarantee the presence of multiple locally competent classifiers rather than just one. As the number of competent base classifiers increases, the probability of selecting only the competent ones should also increase.

IV. Conclusion

In this work, we perform an analysis comparing dynamic selection techniques with the K-NN classifier in order to better understand why and when dynamic selection techniques outperform the K-NN classifier. The analysis is motivated by the fact that the majority of the DS techniques are based on the K-NN definition, and the quality of its neighborhood has a huge impact on the performance of DS methods.

Experimental results demonstrate that the majority of DS techniques obtain a significant improvement in classification performance. Moreover, an analysis conducted using instance hardness shows that the reasons in which DS presents better classification performance is due to the fact that DS techniques are better able to deal with samples with a high degree of instance hardness, while the K-NN classifier works well for samples with a low degree of instance hardness, but fails to predict the correct label for samples with a high degree of IH (the accuracy of the K-NN classifier is close to 0 for samples with an IH of 0.7).

Future work would involve the definition of a system in two steps: first the hardness of a test instance is calculated (based on its neighborhood defined over the training and validation data), and based on its hardness the system could select whether using the K-NN or applying a DS technique for classification. In this case, the DS scheme is only used to classify samples associated with a high degree of instance hardness i.e. borderline samples, while K-NN should be used for classifying samples with a low degree of instance hardness. Such approach would not only improve generalization performance, but also reduce the computational complexity involved, since the DS techniques would only be used for the classification of a few test samples.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the École de Technologie Supérieure (ÉTS Montréal), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and FACEPE (Fundação de Amparo à Ciência e Tecnologia de Pernambuco).

References

[1] R. M. O. Cruz, R. Sabourin, and G. D. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Information Fusion*, vol. 41, pp. 195 – 216, 2018.

[2] A. S. Britto, R. Sabourin, and L. E. S. de Oliveira, "Dynamic selection of classifiers - A comprehensive review," *Pattern Recognition*, vol. 47, no. 11, pp. 3665–3680, 2014.

[3] T. Woloszynski and M. Kurzynski, "A probabilistic model of classifier competence for dynamic ensemble selection," *Pattern Recognition*, vol. 44, pp. 2656–2668, October 2011.

[4] A. H. R. Ko, R. Sabourin, and u. S. Britto, Jr., "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognition*, vol. 41, pp. 1735–1748, May 2008.

[5] R. M. O. Cruz, R. Sabourin, G. D. C. Cavalcanti, and T. I. Ren, "META-DES: A dynamic ensemble selection framework using meta-learning," *Pattern Recognition*, vol. 48, no. 5, pp. 1925–1935, 2015.

[6] X. Zhu, X. Wu, and Y. Yang, "Dynamic classifier selection for effective mining from noisy data streams," *Proceedings of the 4th IEEE International Conference on Data Mining*, pp. 305–312, 2004.

[7] K. Woods, W. P. Kegelmeyer, Jr., and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 19, pp. 405–410, April 1997.

[8] P. C. Smits, "Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 4, pp. 801–813, 2002.

[9] R. G. F. Soares, A. Santana, A. M. P. Canuto, and M. C. P. de Souto, "Using accuracy and diversity to select classifiers to build ensembles," *Proceedings of the International Joint Conference on Neural Networks*, pp. 1310–1316, 2006.

[10] M. Sabourin, A. Mitiche, D. Thomas, and G. Nagy, "Classifier combination for handprinted digit recognition," *International Conference on Document Analysis and Recognition*, pp. 163–166, 1993.

[11] T. Woloszynski, M. Kurzynski, P. Podsiadlo, and G. W. Stachowiak, "A measure of competence based on random classification for dynamic ensemble selection," *Information Fusion*, vol. 13, no. 3, pp. 207–213, 2012.

[12] R. M. O. Cruz, G. D. C. Cavalcanti, and T. I. Ren, "A method for dynamic ensemble selection based on a filter and an adaptive distance to improve the quality of the regions of competence," *Proceedings of the International Joint Conference on Neural Networks*, pp. 1126 – 1133, 2011.

[13] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "A DEEP analysis of the META-DES framework for dynamic selection of ensemble of classifiers," *CoRR*, vol. abs/1509.00825, 2015.

[14] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Automatic recognition of handwritten numerical strings: A recognition and verification strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1438–1454, 2002.

[15] R. M. O. Cruz, R. Sabourin, and G. D. Cavalcanti, "Prototype selection for dynamic classifier and ensemble selection," *Neural Computing and Applications*, pp. 1–11, 2016.

[16] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, vol. 6, no. 1, pp. 63–81, 2005.

[17] E. M. dos Santos, R. Sabourin, and P. Maupin, "Overfitting cautious selection of classifier ensembles with genetic algorithms," *Information Fusion*, vol. 10, no. 2, pp. 150–162, 2009.

[18] G. Giacinto and F. Roli, "Methods for dynamic classifier selection," in *Image Analysis and Processing, 1999. Proceedings. International Conference on*. IEEE, 1999, pp. 659–664.

[19] L. Didaci, G. Giacinto, F. Roli, and G. L. Marcialis, "A study on the performances of dynamic classifier selection based on local accuracy estimation," *Pattern Recognition*, vol. 38, no. 11, pp. 2188–2191, 2005.

[20] G. Giacinto and F. Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognition*, vol. 34, pp. 1879–1881, 2001.

[21] A. L. Brun, A. S. B. Jr., L. S. Oliveira, F. Enembreck, and R. Sabourin, "Contribution of data complexity features on dynamic classifier selection," in *2016 International Joint Conference on Neural Networks*, 2016, pp. 4396–4403.

[22] D. V. Oliveira, G. D. Cavalcanti, and R. Sabourin, "Online pruning of base classifiers for dynamic ensemble selection," *Pattern Recognition*, vol. 72, pp. 44 – 58, 2017.

[23] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "Analyzing different prototype selection techniques for dynamic classifier and ensemble selection," in *International Joint Conference on Neural Networks*, 2017, pp. 3959–3966.

[24] E. M. Dos Santos, R. Sabourin, and P. Maupin, "A dynamic overproduce-and-choose strategy for the selection of classifier ensembles," *Pattern Recognition*, vol. 41, pp. 2993–3009, October 2008.

[25] P. R. Cavalin, R. Sabourin, and C. Y. Suen, "Dynamic selection approaches for multiple classifier systems," *Neural Computing and Applications*, vol. 22, no. 3-4, pp. 673–688, 2013.

[26] S. García, J. Luengo, and F. Herrera, "Dealing with noisy data," in *Data Preprocessing in Data Mining*. Springer, 2015, pp. 107–145.

[27] M. R. Smith, T. R. Martinez, and C. G. Giraud-Carrier, "An instance level analysis of data complexity," *Machine Learning*, vol. 95, no. 2, pp. 225–256, 2014.

[28] T. Woloszynski and M. Kurzynski, "A measure of competence based on randomized reference classifier for dynamic ensemble selection," in *International Conference on Pattern Recognition*, 2010, pp. 4194–4197.

[29] P. R. Cavalin, R. Sabourin, and C. Y. Suen, "Logid: An adaptive framework combining local and global incremental learning for dynamic selection of ensembles of HMMs," *Pattern Recognition*, vol. 45, no. 9, pp. 3544–3556, 2012.

[30] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Meta-des. oracle: Meta-learning and feature selection for dynamic ensemble selection," *Information Fusion*, vol. 38, pp. 84–103, 2017.

[31] M. Friedman, "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, Dec. 1937.

[32] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, Dec. 2006.

APPENDIX II

THE NUMERICAL RESULTS

This appendix contains complementary information to this research. We present three tables providing the numerical results of the 4 DS techniques (Aposteriori, LCA, KNORA-E, KNORA-U) by the different Pool generation approaches, Bagging, SGH and Strategy 4 respectively.

Therefore, the Tables A II-1, A II-2, A II-3 and A II-4 below represent the numerical results of the DS techniques for the different pool generation methods; the mentions B, G and L represent Bagging, SGH and DSLP respectively. The results are given by their mean and standard deviation for the 20 replications. The best results performed by the DS techniques are in bold.

Table A II-5 represents the number of classifiers generated using the (SGH) and the proposed Dynamic Selection Local Pool (DSLP). The number of classifiers generated in DSLP is proportional to the number of hard samples provided by the instance hardness measure.

Classification results with Aposteriori

Table-A II-1 Mean and standard deviation of the generalization accuracy rate of **Aposteriori** with several pools, one of 100 Perceptrons generated using Bagging, the and the proposed pool strategy 4. Best results are in bold.

Datasets	APOS-B	APOS-G	APOS-L
Adult	85.09(0.02)	85.55(0.02)	**87.17(1.71)**
Blood	70.61(0.03)	**71.78(0.02)**	71.25(2.64)
Breast	95.35(0.02)	95.14(0.02)	**95.7(1.46)**
German	69.08(0.01)	69.28(0.01)	**71(1.55)**
Haberman	70.13(0.03)	69.41(0.05)	**72.63(3.64)**
Heart	85.88(0.02)	84.85(0.03)	**86.62(1.84)**
ILPD	67.19(0.01)	66.85(0.01)	**68.29(1.34)**
Ionosphere	80(0.03)	77.27(0.04)	**82.39(3.95)**
Laryngeal1	82.26(0.04)	80.38(0.03)	**82.26(3.02)**
Lithuanian	96.03(0.03)	96.13(0.03)	**96.87(2.37)**
Liver	**64.07(0.05)**	62.09(0.05)	63.49(5.04)
Mammographic	79.11(0.05)	**79.45(0.04)**	78.8(3.71)
Monk2	**83.89(0.06)**	81.67(0.06)	83.43(5.92)
Pima	72.92(0.02)	72.03(0.03)	**75.1(1.97)**
Sonar	**70.87(0.04)**	69.23(0.03)	66.92(3.28)
Vertebral	**82.95(0.04)**	80.77(0.04)	81.92(5.44)
Weaning	73.95(0.04)	72.37(0.03)	**74.47(3.97)**
Average	78.20	77.31	78.72

Classification results with LCA

Table-A II-2 Mean and standard deviation of the generalization accuracy rate of **LCA** with several pools, one of 100 Perceptrons generated using Bagging, the and the proposed pool strategy 4. Best results are in bold.

Datasets	LCA-B	LCA-G	LCA-L
Adult	86.21(0.02)	**87.51(0.02)**	85.55(2.56)
Blood	77.79(0.02)	**78.24(0.02)**	76.22(1.67)
Breast	**96.58(0.01)**	95.42(0.02)	96.13(1.95)
German	**72.58(0.01)**	70.4(0.01)	68.8(1.83)
Haberman	70.79(0.04)	**72.5(0.04)**	69.61(4.5)
Heart	81.18(0.04)	**87.06(0.03)**	79.71(2.51)
ILPD	67.98(0.03)	**69.25(0.02)**	67.88(2.58)
Ionosphere	**87.9(0.03)**	82.27(0.05)	83.3(2.28)
Laryngeal1	79.62(0.03)	**80(0.03)**	79.43(2.8)
Lithuanian	95.73(0.02)	**96.6(0.03)**	96.33(2.21)
Liver	**66.57(0.06)**	61.86(0.04)	64.53(4.57)
Mammographic	**82.79(0.02)**	82.21(0.03)	81.15(1.49)
Monk2	88.06(0.05)	84.44(0.08)	**92.5(4.58)**
Pima	73.05(0.02)	**75.1(0.03)**	72.92(3.37)
Sonar	**78.27(0.04)**	67.12(0.02)	70.96(4.54)
Vertebral	**84.81(0.06)**	80.51(0.05)	83.21(5.24)
Weaning	76.97(0.04)	77.11(0.04)	**81.32(5.12)**
Average	80.40	79.27	79.38

Classification results with KNORA-E

Table-A II-3 Mean and standard deviation of the generalization accuracy rate of **KNORA-E** with several pools, one of 100 Perceptrons generated using Bagging, the and the proposed pool strategy 4. Best results are in bold.

Datasets	**KNORAE-B**	**KNORAE-G**	**KNORAE-L**
Adult	**87.89(0.02)**	86.01(0.03)	86.47(1.92)
Blood	**73.14(0.02)**	72.13(0.03)	72.29(2.13)
Breast	**97.75(0.01)**	94.37(0.01)	96.76(1.03)
German	**73.16(0.02)**	70.52(0.01)	71.44(1.18)
Haberman	**73.55(0.03)**	60.13(0.04)	67.63(6.19)
Heart	**83.68(0.03)**	83.38(0.03)	**83.68(4.45)**
ILPD	**69.14(0.02)**	66.58(0.04)	67.95(2.22)
Ionosphere	**89.49(0.02)**	88.52(0.02)	89.09(2.16)
Laryngeal1	**81.23(0.04)**	78.49(0.02)	79.81(4.16)
Lithuanian	95.77(0.02)	95.8(0.03)	**96.73(2.59)**
Liver	**64.19(0.06)**	57.44(0.05)	56.86(5.27)
Mammographic	**82.14(0.03)**	78.89(0.03)	80.58(2.77)
Monk2	87.87(0.06)	76.76(0.1)	**93.89(6.14)**
Pima	**76.04(0.02)**	67.92(0.03)	75.83(3.8)
Sonar	**84.71(0.04)**	64.81(0.07)	75.38(1.48)
Vertebral	**83.4(0.03)**	79.1(0.04)	81.41(3.01)
Weaning	**81.64(0.02)**	77.63(0.04)	79.08(3.79)
Average	81.46	76.38	79.70

Classification results with KNORA-U

Table-A II-4 Mean and standard deviation of the generalization accuracy rate of **KNORA-U** with several pools, one of 100 Perceptrons generated using Bagging, the and the proposed pool strategy 4. Best results are in bold.

Datasets	**KNORAU-B**	**KNORAU-G**	**KNORAU-L**
Adult	88.7(0.02)	**89.19(0.03)**	88.5(2.27)
Blood	78.4(0.01)	74.63(0.02)	**78.62(1.24)**
Breast	**97.36(0.01)**	95.99(0.01)	97.11(1.63)
German	**76.4(0.02)**	70.72(0.02)	71.68(0.86)
Haberman	**76.12(0.02)**	67.5(0.04)	70.92(4.5)
Heart	**86.62(0.03)**	86.18(0.03)	85.59(3.43)
ILPD	69.11(0.03)	66.71(0.04)	**71.58(1.67)**
Ionosphere	**88.69(0.01)**	88.18(0.02)	84.66(1.9)
Laryngeal1	**84.72(0.04)**	80.19(0.02)	80.75(3.85)
Lithuanian	93.63(0.02)	95.53(0.02)	**96.07(2.51)**
Liver	68.43(0.04)	56.98(0.04)	62.09(5.58)
Mammographic	**84.93(0.03)**	80.72(0.02)	82.16(2.94)
Monk2	83.56(0.06)	80.37(0.07)	**86.57(4.77)**
Pima	77.19(0.02)	72.45(0.02)	**77.81(3.07)**
Sonar	**83.27(0.04)**	67.12(0.04)	68.85(6.29)
Vertebral	**85.32(0.04)**	82.31(0.03)	82.56(4.08)
Weaning	**82.57(0.05)**	78.95(0.03)	74.61(6.67)
Average	82.65	78.45	80.01

Number of classifiers generated

In this part, we present the number of classifiers generated by SGH and the proposed system for several strategies. As strategy 1 relies on the DCS techniques to create DSLP, we kept only the table of APOS. Strategy 2 and strategy 4 have present the same number of classifiers in average for all the DS techniques, since they only keep 1 competent classifier per hard sample. This means that for strategy 2 and 4, the number of classifiers is equal to the number of hard samples. As for strategy 3, it uses KNORA-E to decide which classifiers are kept and added to DSLP, this leads to more than one classifier per hard sample in average. The same conclusion is drawn for strategy 5, since it keeps all the classifiers presenting the highest score of distinction between the pairs of frienemies.

Table A II-5 represents the number of classifiers generated by SGH and the number of classifiers created by the proposed method for strategies 2 and 4. On the other hand, Table A II-6 represents the number of classifiers generated by SGH and the number of classifiers created by the proposed method for strategy 1 for APOS. Moreover, Table A II-7 represents the number of classifiers generated by SGH and the number of classifiers created by the proposed method for strategy 3. Finally, Table A II-8 represents the number of classifiers generated by SGH and the number of classifiers created by the proposed method for strategy 5.

Number of classifiers generated : SGH and strategy 2 and 4

For Table A II-5 represents the number of classifiers generated by SGH and the number of classifiers created by the proposed method for strategies 2 and 4 . The number of classifiers presented by the proposed method is proportional to the number of hard samples found in each dataset, according to the measure of hardness.

Table-A II-5 Mean and standard deviation of the number of classifier provided by the (SGH) and the Dynamic Selection Local Pool (DSLP)

Dataset	SGH	DSLP
Adult	3.1(0.31)	124.20(8.13)
Blood	3.0(0.0)	193.10(11.89)
Breast	3.0(0.0)	43.70(3.47)
German	3.1(0.31)	304.30(8.52)
Haberman	3.8(0.41)	100.00(3.61)
Heart	3.2(0.41)	64.40(8.13)
ILPD	3.8(0.41)	177.80(5.44)
Ionosphere	3.7(0.47)	46.70(7.69)
Laryngeal1	2.4(0.68)	53.30(6.24)
Lithuanian	3.6(0.5)	48.50(7.63)
Liver	3.2(0.41)	126.20(1.58)
Mammographic	2.9(0.31)	194.60(15.87)
Monk2	2.5(0.51)	126.70(3.76)
Pima	3.5(0.51)	999.80(32.43)
Sonar	3.3(0.66)	220.40(6.07)
Vertebral	2.5(0.69)	72.60(2.11)
Weaning	3.0(0.0)	88.60(7.23)
Average	3.15	169.94

Number of classifiers generated : SGH and strategy 1

Table A II-6 represents the number of classifiers generated by SGH and the number of classifiers created by the proposed method for strategy 1 for APOS.

Table-A II-6 Mean and standard deviation of the number of classifier provided by the (SGH) and the Dynamic Selection Local Pool (DSLP)

Dataset	**SGH**	**DSPG**
Adult	3.1(0.31)	66.8(7.01)
Blood	3.0(0.0)	99.1(9.84)
Breast	3.0(0.0)	22.2(6.15)
German	3.1(0.31)	143.2(6.15)
Haberman	3.8(0.41)	53.65(5.65)
Heart	3.2(0.41)	27(9.44)
ILPD	3.8(0.41)	84.1(8.77)
Ionosphere	3.7(0.47)	17.6(5.84)
Laryngeal1	2.4(0.68)	28.4(5.03)
Lithuanian	3.6(0.5)	21.9(7.29)
Liver	3.2(0.41)	61.25(9.59)
Mammographic	2.9(0.31)	84.05(9.56)
Monk2	2.5(0.51)	78.9(3.56)
Pima	3.5(0.51)	113.7(9.52)
Sonar	3.3(0.66)	38.1(10.56)
Vertebral	2.5(0.69)	39(9.26)
Weaning	3.0(0.0)	52(9.51)
Avergae	3.15	132.8

Number of classifiers generated : SGH and strategy 3

Table A II-7 represents the number of classifiers generated by SGH and the number of classifiers created by the proposed method for strategy 3.

Table-A II-7 Mean and standard deviation of the number of classifier provided by the (SGH) and the Dynamic Selection Local Pool (DSLP)

Dataset	SGH	DSPG
Adult	3.1(0.31)	129.10(17.60)
Blood	3.0(0.0)	305.70(28.89)
Breast	3.0(0.0)	56.20(21.10)
German	3.1(0.31)	238.40(25.02)
Haberman	3.8(0.41)	148.20(20.15)
Heart	3.2(0.41)	60.40(14.18)
ILPD	3.8(0.41)	270.10(45.28)
Ionosphere	3.7(0.47)	63.20(11.34)
Laryngeal1	2.4(0.68)	69.80(13.99)
Lithuanian	3.6(0.5)	135.70(36.73)
Liver	3.2(0.41)	141.80(12.20)
Mammographic	2.9(0.31)	403.80(47.75)
Monk2	2.5(0.51)	251.90(35.41)
Pima	3.5(0.51)	226.60(38.70)
Sonar	3.3(0.66)	59.00(15.98)
Vertebral	2.5(0.69)	109.10(25.82)
Weaning	3.0(0.0)	43.00(8.38)
Avergae	3.15	159.52

Number of classifiers generated : SGH and strategy 5

Table A II-8 represents the number of classifiers generated by SGH and the number of classifiers created by the proposed method for strategy 5.

Table-A II-8 Mean and standard deviation of the number of classifier provided by the (SGH) and the Dynamic Selection Local Pool (DSLP)

Dataset	**SGH**	**DSPG**
Adult	3.1(0.31)	454.30(35.65)
Blood	3.0(0.0)	661.20(118.03)
Breast	3.0(0.0)	191.70(13.97)
German	3.1(0.31)	1073.20(20.8)
Haberman	3.8(0.41)	287.70(27.26)
Heart	3.2(0.41)	262.00(16.37)
ILPD	3.8(0.41)	470.90(37.51)
Ionosphere	3.7(0.47)	181.20(33.61)
Laryngeal1	2.4(0.68)	149.00(12.92)
Lithuanian	3.6(0.5)	213.50(39.35)
Liver	3.2(0.41)	343.20(22.04)
Mammographic	2.9(0.31)	726.20(90.14)
Monk2	2.5(0.51)	456.30(23.25)
Pima	3.5(0.51)	648.90(54.20)
Sonar	3.3(0.66)	306.40(49.29)
Vertebral	2.5(0.69)	238.60(27.86)
Weaning	3.0(0.0)	350.20(28.38)
Avergae	3.15	412.61

BIBLIOGRAPHY

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L. & Herrera, F. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17.

Alpaydin, E. (2014). Introduction to machine learning.

Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [Dataset]. Consulted at https://archive.ics.uci.edu/ml/index.php.

Barandiaran, I. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8).

Batista, L., Granger, E. & Sabourin, R. (2011). Dynamic Ensemble Selection for Off-Line Signature Verification. *Multiple Classifier Systems*, (Lecture Notes in Computer Science), 157–166.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2), 123–140.

Britto, A. S., Sabourin, R. & Oliveira, L. E. S. (2014). Dynamic selection of classifiers—A comprehensive review. *Pattern Recognition*, 47(11), 3665–3680. doi: 10.1016/j.patcog.2014.05.003.

Cavalin, P. R., Sabourin, R. & Suen, C. Y. (2012). LoGID: An adaptive framework combining local and global incremental learning for dynamic selection of ensembles of HMMs. *Pattern Recognition*, 45(9), 3544–3556. Consulted at http://www.sciencedirect.com/science/article/pii/S0031320312001124.

Cavalin, P. R., Sabourin, R. & Suen, C. Y. (2013). Dynamic selection approaches for multiple classifier systems. *Neural Computing and Applications*, 22(3), 673–688.

Corne, D. W. & Knowles, J. D. (2003). No Free Lunch and Free Leftovers Theorems for Multi-objective Optimisation Problems. *Evolutionary Multi-Criterion Optimization*, (Lecture Notes in Computer Science), 327–341.

Cruz, R. M. O., Zakane, H. H., Sabourin, R. & Cavalcanti, G. D. C. (2017). Dynamic ensemble selection VS K-NN: Why and when dynamic selection obtains higher classification performance? *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*.

Cruz, R. M. O., Sabourin, R., Cavalcanti, G. D. C. & Ing Ren, T. (2015a). META-DES: A dynamic ensemble selection framework using meta-learning. *Pattern Recognition*, 48(5), 1925–1935. doi: 10.1016/j.patcog.2014.12.003.

Cruz, R. M. O., Sabourin, R. & Cavalcanti, G. D. C. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41, 195–216.

Cruz, R. M. O., Oliveira, D. V. R., Cavalcanti, G. D. C. & Sabourin, R. (2019). FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection. *Pattern Recognition*, 85, 149–160.

Cruz, R. M. (2016). *Dynamic selection of ensemble of classifiers using meta-learning*. (phd, École de technologie supérieure, Montréal).

Cruz, R. M., Cavalcanti, G. D. & Tsang, I. R. (2011). A method for dynamic ensemble selection based on a filter and an adaptive distance to improve the quality of the regions of competence. *IJCNN*, pp. 1126–1133.

Cruz, R. M., Sabourin, R. & Cavalcanti, G. D. (2015b). A DEEP analysis of the META-DES framework for dynamic selection of ensemble of classifiers. *arXiv preprint arXiv:1509.00825*.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7(Jan), 1–30. Consulted at http://www.jmlr.org/papers/v7/demsar06a.html.

Didaci, L. & Giacinto, G. (2004). Dynamic Classifier Selection by Adaptive k-Nearest-Neighbourhood Rule. *Multiple Classifier Systems*, (Lecture Notes in Computer Science), 174–183.

Dos Santos, E. M., Sabourin, R. & Maupin, P. (2008). A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. *Pattern recognition*, 41(10), 2993–3009.

Efron, B. & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York, N.Y.; London: Chapman & Hall.

Freund, Y., Schapire, R. E. et al. (1996). Experiments with a new boosting algorithm. *Icml*, 96, 148–156.

Friedman, J., Hastie, T. & Tibshirani, R. (2001). *The elements of statistical learning*. Springer series in statistics New York, NY, USA:.

Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200), 675–701. doi: 10.1080/01621459.1937.10503522.

Giacinto, G. & Roli, F. (1999). Methods for dynamic classifier selection. *Proceedings 10th International Conference on Image Analysis and Processing*, pp. 659–664.

Giacinto, G. & Roli, F. (2001). Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognition*, 34(9), 1879–1881.

Henniges, P., Granger, E. & Sabourin, R. (2005). Factors of overtraining with fuzzy ARTMAP neural networks. *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, 2005. IJCNN'05*, 2, 1075–1080.

Huang, Y. S. & Suen, C. Y. (1995). A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1), 90–94.

Ivakhnenko, A. (1970). Heuristic self-organization in problems of engineering cybernetics. *Automatica*, 6(2), 207–219.

Jutten, C. (2002). ELENA Project: The enhanced learning for evolutive neural architectures project. Consulted at https://www.elen.ucl.ac.be/neural-nets/Research/Projects/ELENA/elena.htm.

Kaufman, L. & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

KING, R. D., FENG, C. & SUTHERLAND, A. (1995). Statlog: Comparison of Classification Algorithms on Large Real-World Problems. *Applied Artificial Intelligence*, 9(3), 289–333. doi: 10.1080/08839519508945477.

Ko, A. H., Sabourin, R. & Britto Jr, A. S. (2008). From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5), 1718–1731.

Kuncheva, L. (2004a). Ludmila Kuncheva Collection. Consulted at http://pages.bangor.ac.uk/~mas00a/.

Kuncheva, L. I. & Rodriguez, J. J. (2007). Classifier Ensembles with a Random Linear Oracle. *IEEE Transactions on Knowledge and Data Engineering*, 19(4), 500–508.

Kuncheva, L. I. (2002). A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2), 281–286.

Kuncheva, L. I. (2004b). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Levesque, J.-C., Durand, A., Gagne, C. & Sabourin, R. (2012). Multi-objective Evolutionary Optimization for Generating Ensembles of Classifiers in the ROC Space. *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, (GECCO '12), 879–886.

Lustosa Filho, J. A. S., Canuto, A. M. P. & Santiago, R. H. N. (2018). Investigating the impact of selection criteria in dynamic ensemble selection methods. *Expert Systems with Applications*, 106, 141–153.

Mao, S., Jiao, L. C., Xiong, L. & Gou, S. (2011). Greedy optimization classifiers ensemble based on diversity. *Pattern Recognition*, 44(6), 1245–1261.

Oliveira, D. V. R. (2018). *FIRE-DES and FIRE-DES++: Dynamic Classifier Ensemble Selection Frameworks*. (phd, UFPE, Pernambuco). Consulted at https://www.etsmtl.ca/Unites-de-recherche/LIVIA/Recherche-et-innovation/Theses.

Oliveira, D. V. R., Cavalcanti, G. D. C. & Sabourin, R. (2017). Online pruning of base classifiers for Dynamic Ensemble Selection. *Pattern Recognition*, 72, 44–58.

Oliveira, D. V. R., Cavalcanti, G. D. C., Porpino, T. N., Cruz, R. M. O. & Sabourin, R. (2018). K-Nearest Oracles Borderline Dynamic Classifier Ensemble Selection. *International Joint Conference on Neural Networks*, 1804. Consulted at http://adsabs.harvard.edu/abs/2018arXiv180406943O.

Ordóñez, F. J., Ledezma, A. & Sanchis, A. (2008). Genetic Approach for Optimizing Ensembles of Classifiers. *FLAIRS conference*, pp. 89–94.

Ren, Y., Zhang, L. & Suganthan, P. N. (2016). Ensemble classification and regression-recent developments, applications and future directions. *IEEE Comp. Int. Mag.*, 11(1), 41–53.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1), 1–39.

Roy, A., Cruz, R. M., Sabourin, R. & Cavalcanti, G. D. (2016). Meta-regression based pool size prediction scheme for dynamic selection of classifiers. *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 216–221.

Sabourin, M., Mitiche, A., Thomas, D. & Nagy, G. (1993). Classifier combination for hand-printed digit recognition. *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, pp. 163–166. doi: 10.1109/ICDAR.1993.395758.

Santos, E. M. d. & Sabourin, R. (2011). Classifier ensembles optimization guided by population oracle. *2011 IEEE Congress of Evolutionary Computation (CEC)*, pp. 693–698.

Shipp, C. A. & Kuncheva, L. I. (2002). Relationships between combination methods and measures of diversity in combining classifiers. *Information fusion*, 3(2), 135–148.

Skurichina, M. & Duin, R. P. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2), 121–135.

Smith, M. R., Martinez, T. & Giraud-Carrier, C. (2014). An instance level analysis of data complexity. *Machine Learning*, 95(2), 225–256.

Smits, P. C. (2002). Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection. *IEEE Transactions on Geoscience and Remote Sensing*, 40(4), 801–813.

Souza, M. A., Cavalcanti, G. D., Cruz, R. M. & Sabourin, R. (2017). On the characterization of the oracle for dynamic classifier selection. *Neural Networks (IJCNN), 2017 International Joint Conference on*, pp. 332–339.

Souza, M. A., Cavalcanti, G. D. C., Cruz, R. M. O. & Sabourin, R. (2019). Online local pool generation for dynamic classifier selection. *Pattern Recognition*, 85, 132–148.

Surowiecki, J., Silverman, M. P. et al. (2007). The wisdom of crowds. *American Journal of Physics*, 75(2), 190–192.

Tamponi, E. (2015). *Dataset analysis for classifier ensemble enhancement*. (Doctoral Thesis, Universita' degli Studi di Cagliari).

Tang, E. K., Suganthan, P. N. & Yao, X. (2006). An analysis of diversity measures. *Machine learning*, 65(1), 247–271.

Valentini, G. (2005). An experimental bias-variance analysis of SVM ensembles based on resampling techniques. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6), 1252–1271.

Walmsley, F. N., Cavalcanti, G. D. C., Oliveira, D. V. R., Cruz, R. M. O. & Sabourin, R. (2018). An Ensemble Generation Method Based on Instance Hardness. *International Joint Conference on Neural Networks*. Consulted at http://arxiv.org/abs/1804.07419.

Woods, K., Kegelmeyer, W. P. & Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), 405–410.

Woźniak, M., Graña, M. & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16, 3–17. doi: 10.1016/j.inffus.2013.04.006.

Xiao, J. & He, C. (2009). Dynamic classifier ensemble selection based on GMDH. *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on*, 1, 731–734.

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC.